

UNIVERSITY OF OSLO
Department of Informatics

**Towards Topic
Maps for a
Promise Theory
based
Configuration
Management**

Master thesis

Eskedar Kefialew

Network and System
Administration

Oslo University College

Spring 2009



Towards Topic Maps for a Promise Theory based Configuration Management

Eskedar Kefialew

Network and System Administration
Oslo University College

Spring 2009

Abstract

Knowledge management is the major challenge of today. Capturing expert information in a knowledge base that allows non-experts to locate it requires a carefully designed knowledge model. We attempt to discover whether the relationship between Topic Maps and Promise Theory can make configuration knowledge management easier, due to the promise model itself. Topic Maps is an ISO standard for representation and interchange of knowledge. Promise Theory on the other hand is a modeling approach which can be used to model a number of other things including a Policy based configuration management.

Acknowledgment

First, I thank the almighty God for His uncountable support and blessings in my whole life including in this thesis work. Glory to Him?!!!

Secondly, my deep gratefulness and high appreciation goes to my beloved husband, Addisu Tesfaye, for his devotion to keep every aspect of my life on the right track, very comfortable, and so exciting. I also highly appreciate his considerable support in the course of this thesis work. Thank you so much Adye; I always love you so much deeply.

My special appreciation and thankfulness goes to my supervisor, Professor Mark Burgess, for his genuine supervision, encouragement, and invaluable advices and guidance at every stage of this thesis work. Thank you Mark, you have played a very great role for this great achievement of having my Masters in the field which I like most.

I would like to thank the well-known Topic Maps expert, Steve Pepper and his wife for their very useful continued advices and knowledge share, especially, in the area of Topic Maps.

I also would like to thank the whole administration and teaching staff of Oslo University College (OUC) for their extraordinary support and dedication to encourage and help students of the college to achieve their goal. Especially , I have enjoyed astonishing support and help from the head of the computer science department, Associate Professor Thor E Hasle, from the very beginning of my master study in Network and System Administration until today in many areas.

I'm highly grateful to my family, especially, to my mother who has been the main support to let me be always in the line of education from my childhood age.

I would also thank my sister, Tsigareda Asfaw, for her real concern and high support in the course of this thesis work.

Contents

1	Introduction	4
1.1	Concepts behind the problem	4
1.2	Problem statement	5
1.3	Motivation	6
1.4	Summary of Results	6
1.5	Thesis outline	8
2	Background study	9
2.1	Knowledge representation	9
2.2	Knowledge representation technologies	11
2.3	Semantic knowledge representation	12
2.3.1	Resource Description Framework	13
2.4	Topic Maps	16
2.4.1	The TAO of Topic Maps	17
2.4.2	Occurrence Types	18
2.4.3	Association Types	19
2.4.4	Subject Identity	20
2.4.5	Topic Maps Scopes	21
2.4.6	Name Types	23
2.4.7	Topic Maps Constraint and Query Languages	23
2.5	Topic Maps and RDF	24
3	Configuration Management and Promise Theory	27
3.1	Configuration Management	27
3.2	Promise Theory	28
3.3	Cfengine 3: A reference implementation of Promise Theory	29
3.4	Knowledge Management and Promise Theory	31
3.5	Configuration Information	32
3.6	Configuration Management and Knowledge Management: Related Works	33
3.7	Current status of configuration knowledge management	35
4	Design and Implementation	36
4.1	Methodology	36
4.2	Project Start up phase	37
4.3	A Conceptual Model for a Promise Theory based Configuration Management	38

CONTENTS

4.3.1	Inputs of a Promise Theory based Configuration Management domain	38
4.3.2	Processes of a Promise Theory based Configuration Management domain	39
4.3.3	Out puts of a Promise Theory based Configuration Management domain	39
4.4	Domain ontology for a Promise Theory based Configuration Management	41
4.4.1	Topic Types	41
4.4.2	Occurrence Types	43
4.4.3	Association Types	46
4.4.4	Domain Ontology	48
4.5	Implementation of the ontology on Cfengine 3	51
4.5.1	Cfknow and LTM	53
5	Evaluation of Results and Discussion	57
5.1	Evaluation of Results	57
5.2	Discussion	58
6	Conclusions and Future work	60
6.1	Conclusion	60
6.2	Future Work	61

Chapter 1

Introduction

1.1 Concepts behind the problem

Information overload results in low quality of information search. As the number and variety of items involved in a domain increases, human users will have a difficult time in getting the desired information in addition to the problem of viewing the relationships among those items. In this thesis, we consider knowledge in a configuration management domain. The increased number and variety of configuration items such as the different packages and services provided by different machines has been observed to be a serious problem in areas such as big data centers. However, in this thesis work, we have attempted to discover the possibility of solving this problem by using a combination of two approaches namely, Topic Maps and Promise Theory.

Configuration management is a process of establishing and maintaining the right values of configuration parameters in order to yield a human-computer system that conforms to its expected behavior or its Policy state. Promise Theory is a modeling approach that can be used to model a number of other domains including the area of configuration management. In a Promise Theory based configuration management, each component of a human-computer system is expected to make a Promise about its own physical and functional attributes in addition to its relationship with other components. According to the principle of Promise Theory in configuration management, the collective effect of promises made by the different components of a human-computer system is expected to make such systems policy conformant given that the right kinds of promises are made by each component and are continuously maintained by the help of configuration management tools such as Cfengine.

Although there are alternative knowledge representation approaches such as RDF (Resource Description Framework), Topic Maps(TM) is used in this thesis work mainly because of its special relationship with Promise Theory modeling approach. Topic Map's optimality for restructuring of domain information from the perspective of humans and its being an International Organization for Standardization(ISO) standard of knowledge representation are additional factors for the decision of using Topic Maps in this thesis work.

By using Topic Maps to represent the knowledge in a Promise Theory

based configuration management domain, and Promise Theory to model the resulting knowledge structure , we have shown the possibility of using this special relationship for a better management of configuration knowledge .

1.2 Problem statement

Efficient configuration management is very important for an efficient human computer system. Efficiency in the management of the underlying information is one of the major factors that will determine the performance of configuration management. However, as stated in the previous section, the ever increasing number and variety of items involved in configuration management has led to the problem of information overload.

The most highly skilled and intelligent members of society admit difficulty in keeping up with the deluge of new knowledge even in extremely narrow fields[1].

The problem is even more severe in big data centers with thousands of machines leading to poor efficiency of knowledge management. The lack of efficient knowledge management in turn could lead to the specific problems listed here below,

- Results low quality of information search or more specifically it will result in low preciseness and timeliness of information search.
- Viewing the relationships that exist among the different things will be difficult leading to the problem of identifying the cause of some problems.
- Less organized domain information becomes an obstacle for inexperienced people working in the area of configuration management.
- Less organized domain information has low learn-ability leading to the problem of knowledge lose when experienced employees leave organizations.
- A timely reaction towards things such as system failures becomes very difficult.
- Experts will be obliged to waste their time with information search instead of taking care of other advanced tasks.

This thesis attempts to shown a possible way of solving the above listed problems by a semantic reorganization of configuration information by using the standard of Topic Maps.

1.3 Motivation

The lack of integrated configuration knowledge management and the idea of solving it by using the special relationship between Topic Maps and Promise Theory has motivated the authors to work on this problem area.

In addition, the authors were also inspired by an interesting article written by a well known scientist of the 1950s named Vannevar Bush. The article was mainly a prediction made about the future of better information organization for professionals of various fields including lawyers and physicians.

Wholly new forms of encyclopedia will appear, ready made with a mesh of associative trails running through them, ready to be dropped into the memex and there amplified. The lawyer has at his touch the associated opinions and decision of his whole experience, and of the experience of friends and authorities. The patent attorney has on call the millions of issued patents, with familiar trails to every point of his client's interest. The physicians, puzzled by a patient's reactions, strikes the trail established in studying an earlier similar case, and runs rapidly through analogous case histories, with side references to the classics for the pertinent anatomy and histology [2]

The prediction quoted above also works for system administrators or people working in the area of configuration management in general. System administrators, puzzled by the cause of system component failure or confused about the relationship between the different entities will benefit from the realization of this project as a millstone towards a generation of system administration where the life of the professional will be simplified and his efficiency increased through an integrated knowledge management scheme.

People working on the area of configuration management are assumed to be the direct beneficiaries of the result of this research work. That is because, although machines are accomplishing lots of things as they are good at doing what they are told to do, there are things that needs the thinking or analytical skills of human beings. To the knowledge of the authors, the effort of a complete replacement of human beings with machines seems to be far from achievement. That is why the need for increasing knowledge management efficiency from the perspective of humans has become the center of this paper.

The additional benefits that could be gained from solving the specified problems was the other motivating factor. The success of this thesis work could also play a role in solving the problem of interoperability via the resulting Topic Maps among other additional benefits.

1.4 Summary of Results

The broad objective of this study is to discover the possibility having an integrated configuration knowledge management by using the special relationship that exists between Topic Maps and Promise Theory. Figure 1.1 could

1.4. SUMMARY OF RESULTS

represent the big picture of this thesis work. As shown in the figure, in the real world we have places such as big data centers with thousands of machines and other related devices. Keeping track of each device detail such as motherboard serial number or version of installed packages installed on a computer is a very difficult task.

The possibility of using Configuration Management Databases (CMDB) however will alleviate the problem as it could mainly be used as a permanent storage of the underlying information. However, a number of literatures have concluded on the need of higher level knowledge representation approaches such as Topic Maps in order to increase usability of CMDB information. The use of Topic Maps can cover not only the existing configuration management databases but also other information resources such as web pages of some of the device manufacturers. Finally, Promise Theory is used to model the domain knowledge that is represented by Topic Maps.

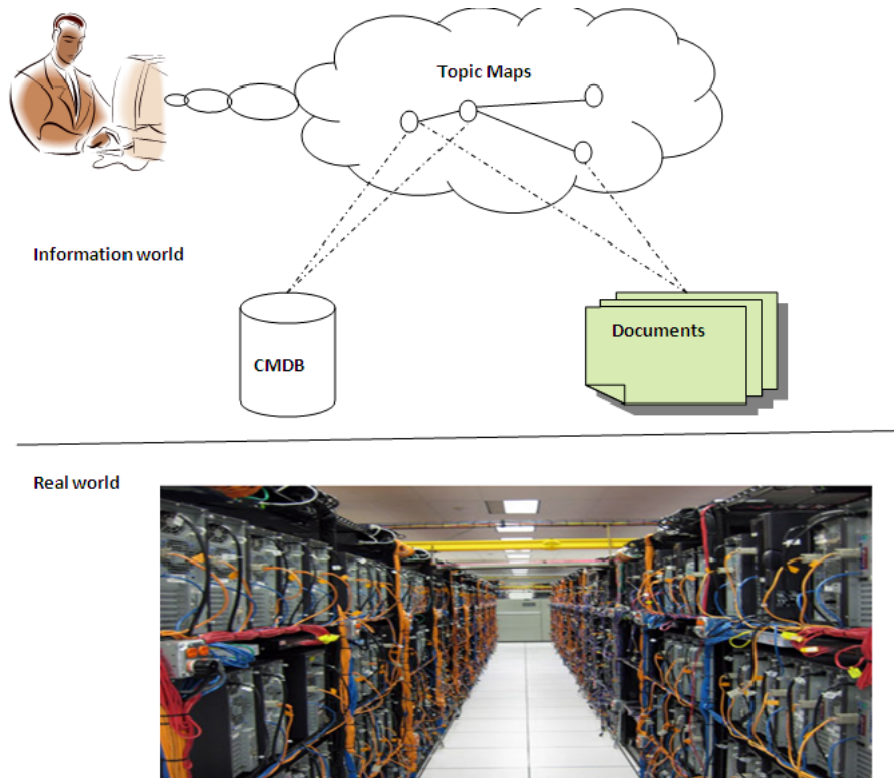


Figure 1.1: Integrated knowledge management based on Topic Maps.

The specific results of this thesis work includes:

- Show the possibility of simplifying configuration information search by using Topic Maps and Promise Theory modeling approaches.
- Show the use of semantically organized domain information as a reasoning aid for the people working on the area of configuration management.

1.5. THESIS OUTLINE

- Show the possibility of minimizing the possible knowledge loss that could happen when experienced employees leave organizations.
- Show the possibility of saving time and enabling timely reaction of different situations by a semantic knowledge organization of a specific domain.
- Show that further management issues such identifying the frequency of system failure in order to know its overall performance will be easier by having an integrated knowledge management scheme.
- Partial ontology for a Promise Theory based configuration management domain is also one of the expected results of this thesis work.

1.5 Thesis outline

- The concept of Knowledge representation and different existing approaches are discussed in following chapter.
- Chapter 3 discusses Configuration Management in general and Promise Theory based configuration management in particular.
- Chapter 4 is about the heuristic methodology adopted in this present work and the details of conceptual design, physical design and implementation phases of the adopted Methodology.
- Chapter 5 is about the result of evaluation followed by discussion.
- Finally, Chapter 6 presents the conclusion and future works that are recommend in relation to this thesis work.

Chapter 2

Background study

The first section of this chapter presents the definition and brief descriptions of the basic concepts that exist in the area of knowledge management. Recently conducted research on evaluating the performance of different knowledge representation approaches by [3] comprises the following section. Among other Knowledge representation approaches, [3] have noted that semantic technologies prove better performance and quality especially in information searches . One of the sections of the chapter deals with the basic ideas behind the two semantic technology giants: RDF and Topic Maps together with the relationship and differences between them. The rationale behind the choice of Topic Maps for this thesis work is also stated in the final section of the chapter.

2.1 Knowledge representation

Today, organizations have to deal with large amount of financial information, personal information of their employees and configuration information of their organization wide computer systems. This large amount of information originating from different sources needs an efficient management in order to keep an organization competent.

Basically, organizational knowledge management involves the acquiring, accessing and maintaining of organizational information. However, all this aspects of knowledge management depends mainly on the way the information is structured. Poor structuring of information therefore will make acquiring, accessing and maintaining of information less efficient. That is why we need a formal way of knowledge representation to increase efficiency of knowledge management.

There are different knowledge representation approaches that are discussed in the following section. Before proceeding to that, definitions of the basic concepts in the area of knowledge management are presented below followed by a brief discussion on the relationships that exist among them. The definitions presented here are specially defined for the purpose of this literature.

Definition 1 *Fact: something which actually exists and its existence could be verified in some way.*

2.1. KNOWLEDGE REPRESENTATION

Definition 2 *Information: communicated fact or facts with a context attached to it in order to be meaningful to the receiver.*

Definition 3 *Knowledge: set of information interpreted in some way to form an understanding.*

Definition 4 *Knowledge representation: a surrogate of a domain of discourse through a formal description of its building blocks in such a way that it could be understood by any one interested.*

Definition 5 *Domain ontology: The concepts with in a given domain and the relationship among those concepts.*

Definition 6 *Meta data: Description of an information resource.*

Definition 7 *Semantic web: an extension of the World Wide Web with an enhanced information management scheme based on meta data.*

Set of communicated facts with some context make up information. That means, facts need special context that will make them meaningful in order to be considered as information by its receiver. Set of information on their turn will make up knowledge after interpretation by their receiver. Knowledge is subjective by its nature due to the fact that the same information can be interpreted differently by different receivers.

The act of fabricating knowledge either for human or machine use is considered to be the meaning of knowledge representation. There are lots of similarities between the concepts of knowledge representation and ontology. Ontology, that originated from the field of Philosophy has a narrower meaning compared to that of knowledge representation. That is because, knowledge representation in addition to the representation of concepts and relationships among them, deals with other things such as the technology that will be used in the process of knowledge fabrication.

Meta data is an old concept that is becoming very popular in relation to its application in enhancing the quality of information search. Among other attributes, a meta data of an information resource can comprise of its creation date, author and so on. Dublin core, one of the main metadata standards used to describe web resources, has fifteen properties in its element sets to describe an information resource. The fifteen properties including the two examples given above are subject to choice as per the need of the user.

The Dublin Core Meta data Element Set is a vocabulary of fifteen properties for use in resource description. The name "Dublin" is due to its origin at a 1995 invitational workshop in Dublin, Ohio; "core" because its elements are broad and generic, usable for describing a wide range of resources[4].

The use of metadata to represent information resources increases the quality of information search in such a way that a document will be searched based on its meaning as documented in the metadata than its syntax.

2.2 Knowledge representation technologies

In [3], the traditional and current knowledge representation approaches have been categorized and compared against a set of criteria including the quality of information search. According to the categorization, the first group of knowledge representation technologies supports content based information search: full text search. That means, the search engine will look for every word in every document in order to find a matching word for a given query. In this category, there is no metadata representation and a very good example is Google, a search engine that has an information retrieval system that depends on keyword matching. Since this type of search is based on syntax rather than semantics, in most cases, it will result in high recall and low precision of search results. In other words a search engine can return high number of less related or unrelated documents in a syntax based search.

The second category of knowledge representation technologies are known for their subject based classification of information resources represented using meta data attributes. For instance, subject attributes are used to classify and retrieve books in libraries.

Conversely, the technologies introduced subsequently, structured search and semantic technologies use flat or hierarchically organized meta data, or meta data organized in a network [3].

Subject based classification is form of classification that is based on the subjects or meanings of the information resources. Controlled vocabularies, taxonomy and thesauri are all traditional techniques that have been in use for long periods to enhance the quality of information search through subject based classification. Controlled vocabulary is a set of terms representing sets of concepts in meta data entries to introduce uniformity and avoid the possible ambiguity that could happen due to the different and subjective assignment of keywords to recourses done usually by authors. The use of controlled vocabularies avoids ambiguity of terminologies by assigning a single term for each unique concept.

Taxonomy is a form of controlled vocabulary where the terms are related to the concepts based on hierarchical relationships.

Taxonomy is subject based classification that arranges the terms in the controlled vocabulary into a hierarchy without doing anything further[5].

Although it has some contribution to the quality of information search, taxonomy lacks much more information about the possible relationships among concepts such as equivalence relationships among synonymous terms other than the mere hierarchies.

That is, thesauri extend taxonomies, by adding more built-in relationships and properties[5].

Thesaurus has better features than taxonomies in a way as there is a possibility to indicate broader and narrower relationships between terms assigned to represent concepts and subjects with in controlled vocabularies. Although thesaurus has more added features than taxonomy, techniques that lie in this category in general are criticized for being structurally fixed for flexible navigation within information resource sets. Another criticism lies on the fact that the classifications process is highly subjective and ambiguous that might also result an erroneous categorization of items. In summary, the second category of knowledge representation techniques make use of subject based classification represented with in a meta data [3].

The last category of knowledge representation approaches that comprises Topic Maps and RDF are known for a network like organization of meta data to enable semantic based representation of knowledge. The idea of semantic based knowledge representation is directly related with the idea of the Semantic Web as discussed in the following section.

The result of the evaluation conducted in [3] especially in the quality of search and integration capability of heterogeneous information, the semantic knowledge representation technologies have shown a higher performance. That is why a semantic knowledge representation approach is preferred to increase find ability of information in a Promise Theory based configuration management domain.

Before going to the next section, the author would like to emphasis the fact that by using semantic knowledge representation technologies over the full text search and traditional approaches, the quality of information search will be improved as proved in [3]. The following section discusses the basic ideas behind the Semantic Web initiation and the major existing semantic knowledge representation technologies including the one used in this project.

2.3 Semantic knowledge representation

In the 1950s, the world was moving towards using computers, whereas starting from 2000, the shift has become towards using a world wide connected web of information. In connection with this, there is a paradigm shift towards digitalization of all information resources in order to avoid physical limitations to information access. However, as the size of the World Wide Web (WWW) increases, the traditional approach of web information management is found to be less scalable. That is when the Semantic Web initiation came to the scene with the goal of enabling efficient knowledge management for the WWW through semantic representation of information resources. The term semantic is defined as something relating to meaning in the Oxford dictionary of English language.

The realization of Semantic Web is believed to bring the following advantages among many others as stated in [6].

- Summarized but important information search results will be possible
- Automated reasoning could be achieved

2.3. SEMANTIC KNOWLEDGE REPRESENTATION

- Knowledge will be organized in conceptual space according to its meaning
- Query answering over several documents will be supported
- Semantic interoperability will be possible through ontology mapping
- The accuracy of web searches will be improved including the problem of missing important information.
- Differences in terminologies will be resolved using standard abstract data models.

The goal of Semantic Web is to enhance the quality of global knowledge sharing. In order to achieve this goal, each information source owner is expected to reorganize his/her information resources according to the standards of semantic knowledge representation technologies. This is believed to be a challenge in the realization of the Semantic Web initiation. However, the semantic restructuring of every information resource including those that will be used only locally will enhance the quality of knowledge management.

For a semantic representation of domain knowledge, semantic technologies are required in order to encode meanings independently of the underlying syntax. This will enable machines as well as people to understand, share and reason semantically. Using semantic technologies, adding, changing and implementing new relationships can be highly simplified compared to the traditional approaches. In traditional knowledge representation systems, meaning and relationships must be predefined. This will make them less flexible for possible changes.

The two well known semantic technologies RDF and Topic Maps have the same goal of tackling the fundamental problems of information find-ability and semantic interoperability. These two semantic technology giants have originated from two famous international-standard-setting organizations namely, W3C (World Wide Web Consortium) and ISO (International Organization for Standardization). W3C recommends RDF for semantic web representation where as ISO has set Topic Maps as a standard for knowledge representation and interchange.

2.3.1 Resource Description Framework

RDF comprises set of technologies organized into layers to achieve the implementation of the Semantic Web. The different technologies in these layers work with each other in a harmony towards achieving their common goal. This section discusses the major technologies that comprises the RDF model.

The bottom layer of RDF model as presented in figure 2.1 comprises the Uniform Resource Identifier (URI) and Unicode aspects of knowledge representation. The role played by URIs is explained by [7] as quoted below.

The subject of every assertion (or statement) in an RDF model is a resource, identified by a URI. The subject of every assertion in

2.3. SEMANTIC KNOWLEDGE REPRESENTATION

a topic map is a topic, representing a subject, which may be addressable or non-addressable. Addressable subjects are identified by their URIs (as in RDF); non-addressable subjects are identified by the URIs of (one or more) subject indicators.

Hyper Text Markup Language(HTML) is good at visualizing information for human users. But, when it comes to meaning representation or semantics of data, it is very poor. That is why Extensible Markup Language(XML) has been made part of the XML layer in RDF models. XML is a language enabling web users to use their vocabulary for a better semantic representation of web contents. Today, HTML and XML work together to give a better visual and semantic representation of web contents.

The layer above the XML layer, the RDF layer consists of a language, namely RDF that is used to represent knowledge in a machine understandable format for simple reasoning and other similar purposes. XML represents meta data or information about information. On the other hand RDF represents the information itself. RDF has vocabularies used for making simple sentences in the domain ontology as opposed to the normal language that we use for everyday life (English). On the same layer with RDF, there is RDF Schema containing the list of RDF vocabularies and their descriptions.

RDF was originally created in 1999 as a standard on top of XML for encoding meta data-literally, data about data. Meta data is, of course, things like who authored a web page, what date a blog entry was published, etc., information that is in some sense secondary to some other content already on the regular web. Since then, and perhaps especially after the updated RDF spec in 2004, the scope of RDF has really evolved into something greater. The most exciting uses of RDF aren't in encoding information about web resources, but information about and relations between things in the real world: people, places, concepts, etc [8]

RDF is an ontology representation language. However, it lacks reasoning capability and also has limited vocabularies for better representation of domain ontology. These problems are solved by its follower named the Web Ontology Language (OWL). OWL is built up on RDF and has more vocabularies as a language in addition to its reasoning capability. The ontology language layer is the layer comprising powerful ontology languages such as OWL.

The logic, proof and trust layers of RDF model are found at the top of the layer stack. Logic is a formal language for expressing knowledge with correct reasoning. On the logic layer, logic will help automated reasoners to deduce conclusions from the given knowledge, thus making implicit knowledge explicit. Actual deductive process as well as representation of proofs in Web languages is the role of Proof layer. The trust layer works on ensuring user's trust on the operation conducted by lower layers of the model.

The diagram presented below is found in [6] and although this architecture is debatable and subject to change in the future, it can show the over all structure of the RDF layered approach discussed above.

The Semantic Web Layer Tower

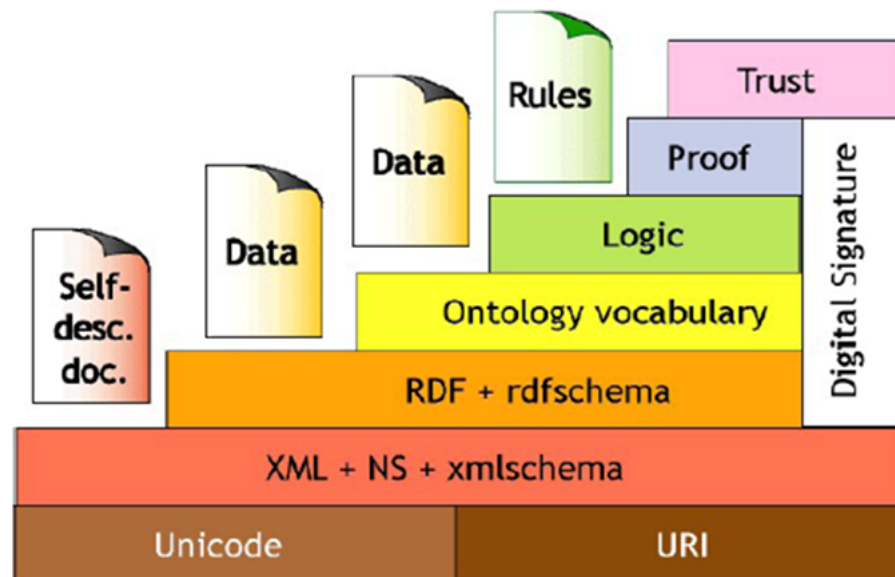


Figure 2.1: RDF/Semantic Web layer cake

2.4 Topic Maps

The idea of Semantic Web could be implemented either through the W3C recommendation of RDF modeling approach or else through the use of Topic Maps. Topic Maps has been set as the ISO standard for knowledge representation and interchange since 2000 and is comprised of nine standards. The ISO standard is formally known as ISO/IEC 13250:2000. Topic Maps is known for its emphasis on find-ability of information by its human users.

The recent move towards semantic technologies has resulted in two standards for interchanging semantic information; RDF and TM. The biggest difference is that Topic Maps approach knowledge representation from the point of view of human being, whereas RDF takes a machine approach [9].

Before going to the detailed discussion of Topic Maps, its relation ships with navigational aids such as indexes, glossaries and thesauri needs to be clarified as today's Topic Maps is constructed based on the good features of these related technologies.

Indexes, glossaries and thesauri are all ways of mapping the knowledge structures that exist implicitly in books and other sources of information.[10]

A typical index constructs are the same as that of Topic Maps as pointed out in [10], this is because, both of them are made up of set of topics, associations and occurrences. However, Topic Maps has some more features such as typing of topics, associations, association roles, names and scoping of topics.

The key features of a typical index are thus: topics (identified by their names, of which there may be more than one); associations between topics; and occurrences of topics (pointed to via locators). For each of these constructs it is useful to be able to say something about the type, in order to convey more information to the user. Topics, Associations and Occurrences are also the key constructs in the topic map model (hence the title of this paper)[10].

The Oxford dictionary of English language defines glossary as an alphabetical list of words related to a specific subject and their explanations. Although a glossary has many features in common with topic maps, the fact that it has only one occurrence type which is the definition of topics has made it less usable among many other missing features that are available in topic maps.

Thesaurus is also a Topic Maps related technology.

Given a particular term, a thesaurus will indicate which other terms mean the same, which terms denote a broader category of the same kind of thing, which denote a narrower category, and which are related in some other way.[10]

2.4. TOPIC MAPS

The key or emphasized feature of a thesaurus is the associations between terms as opposed to other technologies such as indexes and glossaries. The special thing about associations within a thesaurus is the fact that they are typed. Rather than simply saying that there is an association between two terms, the type of association such as a term is broader term than the other one is specified in a thesaurus. However, its fixed structure has made it to be surpassed by Topic Maps as quoted below.

The result is an information structure that breaks out of the traditional hierarchical straight jacket that we have gotten used to squeezing our information into. A topic map usually contains several overlapping hierarchies which are rich with semantic cross-links like "Part X is critical to procedure V." This makes information much easier to find because you no longer act as the designers expected you to; there are multiple redundant navigation paths that will lead you to the same answer. You can even use searches to jump to a good starting point for navigation[11].

The reader is also referred to section 2.2 to have an overview of the performance of these related technologies especially on the quality of information search as compared to the technology chosen for this work, namely Topic Maps.

2.4.1 The TAO of Topic Maps

Topics, Associations and Occurrences (TAO) are the basic constructs of Topic Maps. A topic is a representation of a subject in Topic Maps. Due to its subject centricity, topics: the information world equivalents of real world subjects are at the heart of Topic Maps.

What then is a topic? A topic, in its most generic sense, can be any thing, whatsoever, a person, an entity, a concept, really anything, regardless of whether it exists or has any other specific characteristics, about which anything whatsoever may be asserted by any means whatsoever. [10]

Therefore, for every real world subject of interest, a topic will be defined in the Topic Maps information world. The definition of a topic will assert the existence of that specific subject in a one to one relationship. A topic could be any thing about which one would like to say some thing including things that do not exist.

Similar topics could be categorized under a topic type. Topics in one topic type should have one or more similar things to bring them into one. A topic type could have zero or more instances as one could create a topic type for a non existent thing. Topic types are defined as topics as every thing else in the Topic Maps based information structuring.

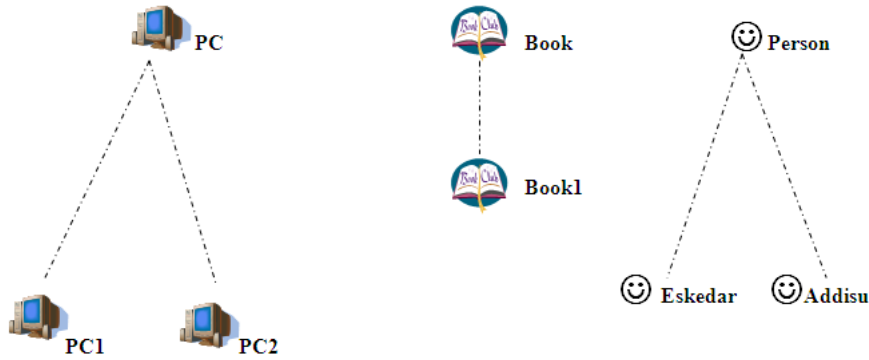


Figure 2.2: The relationship between topics and topic types

In figure 2.2, three topic types are presented. These topic types are PC(Personal Computer), Book and Person. The second topic type, Book has only one instances topic called Book1 where as each of the other two topic types have two topics as their instances.

2.4.2 Occurrence Types

An occurrence is an information resource of a topic. As that of similar topics , similar occurrences will form an occurrence type. An occurrence type of a topic could be either its properties or an external information resource such as a web page that are believed to have relevance to the topic. As presented in following sub section below, occurrence types are classified into two as internal occurrences and external occurrences.

A topic may be linked to one or more information resources that are deemed to be relevant to the topic in some way. Such resources are called occurrences of the topic[10].

Unlike in other modeling paradigms such as database design, in Topic Maps, different topic types can have the same occurrence types. In a database design, each table(topic type equivalent of topic maps) has a set of properties(equivalent of occurrences in topic maps) specifically defined for itself rather than using an independently defined set of occurrence types as it is in the case of Topic Maps.

An occurrence in Topic Maps can have occurrence role and occurrence role type. Although currently the term occurrence role type is changed to occurrence type in order to avoid possible confusions, it may help to say few words about the difference between these two concepts. An occurrence role tells about the nature of the occurrence by stating whether it is a web page or an article written about the topic. The occurrence role type on the other hand explains the relevance of that information resource to the subject in question by stating whether the document defines the topic or only mentions it in relation to some thing else.

Internal Occurrences

The internal occurrences are equivalent to the properties of a topic with the exception of name property as it has a special significance in the design of Topic Maps ontology. An attribute of a relational database table is the equivalent of a Topic Maps internal occurrence. The internal occurrences of a topic have specific data types for the values they assume such as integer for the age property of a person. In Topic Maps, if a property of a topic can take more than one value, the recommended approach is to create a topic for the property and to link it with the original topic for which it is a property for. For example in this project, the mother board serial number of a machine can assume only a single value of string X, whereas the same machine can have multiple Network Interface Cards (NIC). Therefore, NIC as a multi value property is defined as a topic by itself and an association is used instead of using occurrence type to connect with the topic it is a property for. The mother board serial number on the other hand is connected to the topic using occurrence type as it is a single valued attribute.

External Occurrences

External occurrences include any external information resource that could say something about the topic in question such as a web page. External occurrences have Uniform Resource Identifier (URI) as their data types.

Such occurrences are generally external to the topic map document itself (although they may also be inside it), and they are "pointed at" using whatever mechanisms the system supports, typically URIs in XTM.[10]

Topic maps have two ways of external occurrence representations. These are either defining the external occurrence as a topic by itself and associating it with the topic it talks about or to define it as an external occurrence in the same way as one defines an internal occurrences. The rule of thumb as stated in [10] is to create a topic if there is a need to say something about the resource and to define it as external occurrences if that is not the case. For example, if a person has a web page as an information resource, one way of representation is to define a specific topic for that web page if there is an interest in saying something about the web page itself. The author and the creation date are among the things that could be said about the web page if there is any need for it. However, if there is no need for that type of information about the information resource, the web page could simply be defined as an external occurrence type in a similar way as one defines an internal occurrences type.

2.4.3 Association Types

A relationship between two or more topics is asserted by the existence of an association. Like that of topics and occurrences, similar association will also be categorized to form an association type. In addition, each topic involved in an

2.4. TOPIC MAPS

association has a role to play. Similar roles on their turn will form role types. The idea of typing is claimed to be the strength of topic maps for semantic data representation.

The ability to do typing of topic associations greatly increases the expressive power of the topic map, making it possible to group together the set of topics that have the same relationship to any given topic. This is of great importance in providing intuitive and user-friendly interfaces for navigating large pools of information[10].

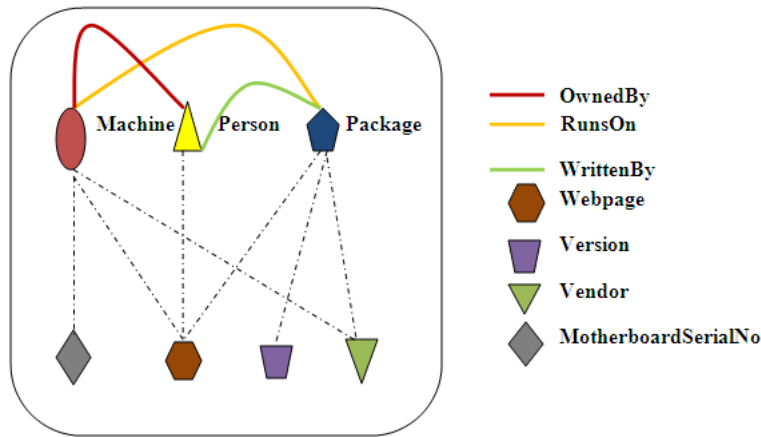


Figure 2.3: Topic types, Occurrence types and Association types

The diagram presented above shows the relationship between the three major Topic Maps constructs discussed above: topics, occurrences and association. In the diagram, three topic types: person, machine and package are connected by three types of associations, namely, written by, owned by and runs on. The only external occurrence type presented in the diagram, web page serves as an external information resource for all of the topic types. The three properties version, vendor and mother board serial number are defined as internal occurrences of the respective topic types.

2.4.4 Subject Identity

If a Topic Map is isolated, subject identity is often unambiguous. However, when it comes to global information sharing as in the case of Semantic Web, a subject needs identification to avoid possible ambiguity. Mostly, subjects could be identified by their names. However, the existence of things such as homonyms and synonyms could create ambiguity in using names for identification. Human beings can solve such problems by using contexts in addition to names. Machines on the other hand are not naturally capable of contextualization. That is why machines need subject identifier: a string that uniquely identifies one subject from the others. A Subject identifier could be local for

2.4. TOPIC MAPS

isolated information resources such as organizational databases or it could also be global so that it can uniquely identify the subject globally.

The goal with topic maps is to achieve a one-to-one relationship between topics and the subjects that they represent, in order to ensure that all knowledge about a particular subject can be accessed via a single topic. However, sometimes the same subject is represented by more than one topic, especially when two topic maps are being merged. In such a situation it is necessary to have some way of establishing the identity between seemingly disparate topics. For example, if reference works publishers from Norway, France and Germany were to merge their topic maps, there would be a need to be able to assert that the topics 'Italia', 'Italie' and 'Italien' all refer to the same subject. The concept that enables this is that of subject identity[10].

A subject is identified by subject locator if the topic is an information resource like a web page or by subject identifier if the topic is not an information resource. A subject identifier is the address of an information resource which says something about a non information resource subject. Such kind of documents are called subject descriptors and they are mainly used for human understanding of the subject in question where as the subject identifiers are mainly used by machines for the purpose of identifying the subject[10].

2.4.5 Topic Maps Scopes

A topic has name, occurrences, associations and roles as its characteristics. These characteristics need to be defined in order to assert the existence of the topic. The definition of these characteristics some times leads to a problem of ambiguity that is when we need scopes mainly for the purpose of disambiguation. In addition to disambiguating topic characteristics, scopes will help in the redesigning of a Topic Map according to some additional views. That means, scopes can be added to increase the no of ways one can interpret a given topic if required for some purpose. One possible misconception regarding scopes is the belief that 'topics have scope'. However, it is not the topic that has a defined scope rather its characteristics that will have scopes defined for their validity.

Scope determines the validity of topic characteristics. In principle, a characteristic of a topic has one or more scopes. That means a topic characteristic such as a role played by a topic at least will have a universal validity if there is no other explicitly defined scope for the same purpose. For example, a person may play the role of being a student in a scope called education; where as the same person can play the role of being a citizen in a scope called country.

Occurrence as a topic characteristic is one source of ambiguity. For instance, if all of the topics within a topic type do not have the same set of characteristics, then there is a need of disambiguation to identify which topics of that specific topic type are having those ambiguous characteristics and which

2.4. TOPIC MAPS

do not such as not all persons have the property of being an author. Therefore, there might be a need for defining a scope with a list of authors to identify the people who are having that specific property of authorship. The same is the case with roles played by a topic in associations.

There should be a scope that determines the validity of a role played by a topic in a given association. One useful and potentially very powerful application of scope is to permit the capture of different "Weltanschauungen", or world views, of the subject. This is extremely important when merging topic maps, since it permits knowledge of which assertions came from which source to be retained: The individual names, occurrences, and associations can be scoped in such a way as to indicate where they originated. However, this is not the only application of scope, as we shall see[12].

Name is the other topic characteristics which becomes a major source of ambiguity in topic map based information restructuring. The two most common problems in relation to topic names are synonyms and homonyms. Synonym is when we have a topic that could be called by different names. In the picture presented below, a topic called Netherlands is referred by two names called Netherlands and Holland. The second case, homonym is when the same name is used by more than one topic in this case, the name Cfengine is used by the company Cfengine and the Package Cfengine. Synonym as a problem could be solved by defining scopes. The problem caused by the existence of homonym on the other hand could be solved by the help of topic types instead of scopes.

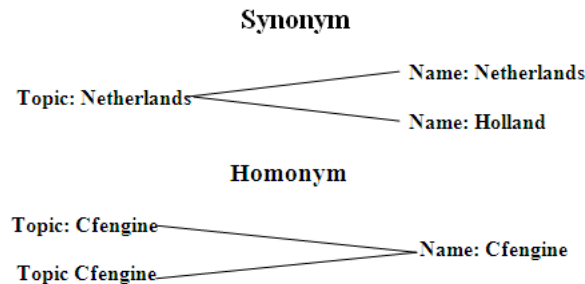


Figure 2.4: Homonyms and Synonyms

Scope has direct relationship with Topic Maps naming constraint that state the impossibility of having same base name for two topics with in a given scope. Even if they do exist, they are supposed to merge as one topic according to the principle of merging Topic Maps. In other words, the definition of scopes will establish a name space in which case no two topics will have the same base name with in the same topic map scope.

As quoted below, Topic Map Scope is defined by a set of topics.

According to the 13250 definition, scope is "the union of the subjects of the set of themes used to specify that scope". NOTE 3 un-

derlines this by making clear that in order to express the intersection of two topics, a new topic must be created. Thus a scope constituted by the themes "history" and "economics" covers the sum total of both of those subject domains. To describe their intersection (i.e., a single domain that has both a history component and an economics component), a new topic must be created (e.g., "economic history" or "history of economics") [12].

The ambiguity problem with topic names could also be solved by the help of typing names as presented in the following section. The choice of scope or typing names depends on some factors including the topic map tools that will be used for the implementation. For instance, Linear Topic Map Notation (LTM) doesn't support typed names whereas Ontopoly on the opposite doesn't currently support scoped names.

2.4.6 Name Types

Name obviously is one of the properties of a topic but, it has its own special role in topic maps ontology design that makes it unique from other properties of a topic. A name is the default characteristic to identify an entity from the other. As stated in the previous section, a name could be source of ambiguity for a number of reasons including the possibility of having homonyms and synonyms. In topic maps, the problem of synonyms could be addressed either through the use of scopes as it is discussed in the previous section or via typing names as presented in this section.

A topic name type is a subject describing the nature of the topic names of that type. Topic names always have a scope, which defines in what context the topic name is an appropriate label for the subject. A topic may have any number of topic names. A base name is a name or label for a subject, expressed as a string. That is, it is something that identifies the subject (though not necessarily uniquely) and can be used as a label for the subject in user interfaces. The notion of a base name corresponds closely to the common sense notion of a name [13].

Topic names could be typed as first name, last name, nick name and given name for a topic of type person. The rule of thumb regarding defining a name as a name type is to see if that name type is used by all members of the topic type. For example, if all students of a school have gotten user name to have access to Information Technology (IT) infrastructure, user name can be defined as a name type to be used by members of the topic type called person. The definition of name types will avoid the possible ambiguity that could happen when different users use different names to search for the same topic.

2.4.7 Topic Maps Constraint and Query Languages

Topic Maps Constraint Language (TMCL) ensures the validity of a designed Topic Map according to the Topic Maps standard or Topic Map data model.

TMCL specifies when a topic, an occurrence, an association or a topic name with in a designed Topic Map will be considered as valid. Topic maps naming constraint for example states the impossibility of having same base name for two topics with in the same scope. Therefore, the constraint language will insure the proper representation of domain knowledge according to the standards of Topic Maps.

Topic Map Query Language (TMQL) is a query language that is used to retrieve information resource from Topic Map based domain ontology. Ontopia has defined a query language called Tolog which is being used by some Topic Maps development tools such as the OKS Sampler. However, TMQL is under the process of standardization by ISO to be the standard query language of the Topic Maps family.

2.5 Topic Maps and RDF

Although they look a complete rivals, RDF and Topic Maps are considered to be complementary technologies as to the literature quoted below. The success of their interoperability effort by Ontopia, is one of the signs for the validity of this claim. However, the interoperability was made by Ontopia; a company dedicated to providing information and knowledge management solutions based on the topic map (XTM) standard instead of an independent organization that could have increased the trustworthiness of result.

Topic maps and RDF originate from two standards organizations, ISO and the W3C respectively, that have traditionally been regarded as competitors. This accounts to some extent for the tendency among the uninformed to regard topic maps and RDF as competitors. Our position is that it makes more sense to regard topic maps and RDF as complementary, and to look for ways of realizing the potential synergies between the two. Ontopia has clearly demonstrated this potential through its use of RDF (under the covers, as it were) in the automated generation of topic maps[7].

RDF and Topic Maps have a number of similarities and differences. They both have semantic view of information structuring and use XML for meta data representation. However, as they are two technologies originating from different application areas and authorities (ISO and W3C) , they exhibit some differences such as the way they define constraints and query languages. RDF has its root in formal logic where as Topic Maps has roots in traditional finding and indexing aids such as back of book indexes, glossaries and thesaurus.

Topic mapping has its roots in traditional finding aids such as back-of-book indexes, glossaries and thesauri. RDF has its roots in formal logic and mathematical graph theory. Topic mapping is knowledge representation applied to information management from the perspective of humans. RDF is knowledge representation applied to information management from the perspective of machines. This accounts for some of the critical differences between the two[7].

2.5. TOPIC MAPS AND RDF

The way information is structured in topic maps is much similar to ours, human beings', way of managing information. This feature makes Topic Maps intuitive knowledge representation technology more suitable when targeting a human audiences. For instance, human beings manage information by categorization and using scopes to contextualize in order to avoid possible ambiguities. Among other things, this makes Topic map way of information restructuring more similar to it.

Topic Maps is very applicable in collaborative solutions because they resemble the way users think[14].

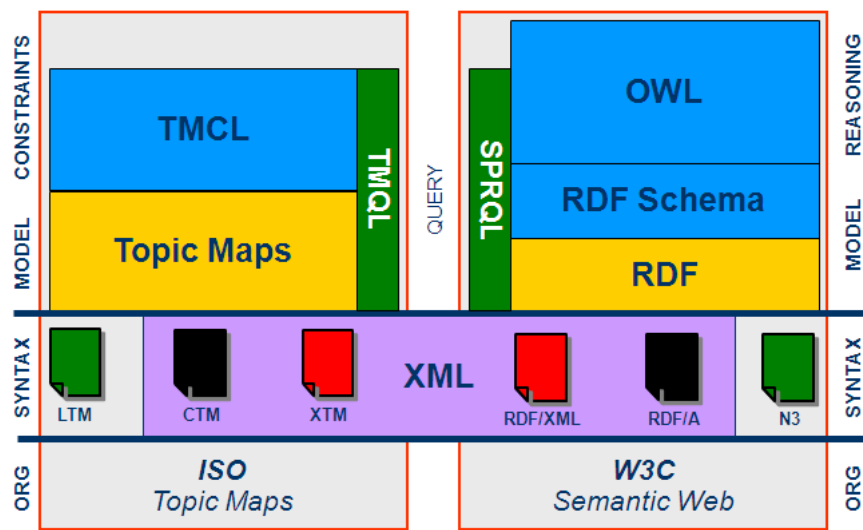


Figure 2.5: Equivalence of the RDF and Topic Map families

The bottom layer of Figure 2.5. shows the two standard-setting-organizations from which RDF and Topic Maps have originated from. The second layer shows the different textual notations that can be used to encode the constructs of Topic maps and RDF standards. Linear Topic Map notation (LTM) represents topic map contracts just like XML Topic Maps (XTM) which is the XML syntax for Topic Maps. XML as stated in the previous sections, is a markup language that enables users to use their vocabulary for a better semantic representation of web continents. RDF/XML is XML syntax for RDF. RDF/A and N3 are also textual notations used to encode the standards of RDF model.

Topic Map Query Language and SPARQL are the query languages used by the two models. Topic Map Constraint language is used to ensure the constraints and schemas in the ontology of developed Topic Map models. Web Ontology Language(OWL) is one of the technologies in the RDF model with a set of vocabularies to serve as language to represent an ontology and allow reasoning capabilities.

In addition to its optimality for restructuring of information from the perspective of humans, the reasons why Topic Maps are chosen as a knowledge representation approach in this thesis work are listed here below.

2.5. TOPIC MAPS AND RDF

- It is an ISO standard for knowledge representation and interchange with emphasis on the find-ability of information.
- It has a special relationship with Promise Theory(See section 3.4)
- It enables seamless navigation through information domain.
- It enables precision on information search and retrieval.
- There are enough and capable tools available that makes its implementation much easier
- It has good support for full text searches and complex queries
- It enables additional view of domain information through visualization

Chapter 3

Configuration Management and Promise Theory

This chapter discusses the meaning of configuration management in general and a Promise Theory based configuration management in particular. A discussion on the Promise Theory modeling approach and its reference implementation namely, Cfengine3 is also included. There is also a section that discusses the meaning of configuration information. Finally the last part of the chapter presents the related works done on the area of knowledge management and configuration management.

3.1 Configuration Management

As stated in [15], operational management of human-computer systems is at the heart of network and system administration. As a system involving humans and computers, both the technology of computer systems and the users of the technology should be addressed in order to have the desired kind of behavior from such systems.

The process of initial setup and continuous follow up of such systems with the goal of ensuring their desired behavior is what we call it configuration management. Configuration is defined by [16], as follows.

Configuration management is the process of constraining the behavior of a network of machines so that each machine's behavior conforms to predefined policies and guidelines and accomplishes predetermined business objectives.

In this thesis work, a configuration management system is taken to be a system involving people, zero or more configuration management tools such as Cfengine and set of interconnected computers with a set of configuration processes intended to give a policy conformant system as an output. A policy conformant system is a system of interconnected computers with the right values for all or at least most of the configuration parameters so that it behaves

3.2. PROMISE THEORY

as expected on the initial policy specification of the system. Configuration parameters are properties of computers that one can assign the desired values to determine the configuration status of the computer system. Examples of configuration parameters include permission of a file, address of a network interface card or the type of file system that could be mounted on a disk volume.

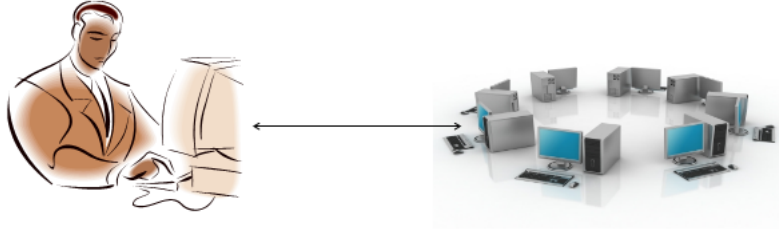


Figure 3.1: A simple configuration management system with a configuration personnel and network of interrelated machines.

The definition we shall use here is as follows:

Definition 8 *Configuration Management: is the process of assigning the desired values to systems configuration parameters and a continuous look up over the behavior of the system in order to make sure that it has an acceptable level of performance.*

3.2 Promise Theory

Promise Theory is a modeling approach for system co-operation introduced at Oslo University College. It can be used to model a number of other things including the domain of configuration management[17]. Promises are at the center of Promise Theory.

A promise is an announcement of fact or behavior by a promiser to a promisee, observed by a number of witnesses (referred to as the scope of the promise), whose outcome has yet to be assessed[18].

[19], has categorized the different kinds of Promises into two special types as presented below.

- A promise to agree to behave like another: is essential for defining groups, roles and social structures with consensus behavior.
- A Promise to utilize the promise of another: is crucial for client-server interaction, dependencies and access control.

The model world such as configuration management in which we formulate promises is expected to have the following characteristics and a summary of notations that could be used are presented in figure 3.2

3.3. CFENGINE 3: A REFERENCE IMPLEMENTATION OF PROMISE THEORY

- There must be agents in order for promises to exist
- There must be a promiser(or source agent)
- There must be a promisee (or receiving agent) which might be the same as the source.
- There must be a body which describes the nature of the promise.

Symbol	Interpretation
$a \xrightarrow{+b} a'$	Promise with body b
$a' \xrightarrow{-b} a$	Promise to accept b
$v_a(a \xrightarrow{+b} a')$	The value of promise to a
$v_{a'}(a \xrightarrow{+b} a')$	The value of promise to a'

Figure 3.2: Summary of Promise Theory notations

The use of Promise Theory for modeling configuration management domains has found to be successful through its reference implementation as discussed below.

3.3 Cfengine 3: A reference implementation of Promise Theory

There are two well known configuration management approaches; Centralized and Policy based management. In both cases, set of specifications about the behavior of each computer will be prepared centrally to determine the expected behavior of the whole system. The Centralized management approach will forcefully push the centrally prepared rules and regulations with or without the will of the end user hosts.

The Policy based management framework such as in Simple Network Management Protocol (SNMP) on the other hand gives autonomy to the agents in a way that they have the full right to pull and implement a centrally prepared set of configuration policies. The possibility of using Promise Theory to model the autonomy of agents in a Policy based configuration management was shown through the configuration management tool Cfengine 3. In Promise Theory, every autonomous agent will make a Promise about its expected behavior based on its choice. That is what makes Promise Theory optimal for a Policy based management framework.

The theory of promises describes policy governed services, in a framework of completely autonomous agents, which assist one another by voluntary cooperation alone[20].

3.3. CFENGINE 3: A REFERENCE IMPLEMENTATION OF PROMISE THEORY

Cfengine 3 is a configuration management tool that has used Promise Theory as its underlying modeling approach. In Cfengine3, as a Promise Theory based configuration management tool, every configuration item will make a promise about its own characteristics and its relationship with other configuration items.

A promise is a specification of future state or behavior from one autonomous agent to another. It is thus a unit of policy. [16]

In this literature, a Configuration Item(CI) is a fundamental structural unit of computer system that needs some sort of setup to serve its intended purpose. If all or at least the major configuration items makes and maintains the right set of promises, the whole system is believed to be a policy conformant system. The different promises that could be made in Cfengine 3 revolve around the major configuration items forming different Promise types. The major Promise types that could be made and the grammatical form of a Cfengine 3 Promise is presented here below;

Promise Types	Grammatical form of a Cfengine 3 Promise
<ul style="list-style-type: none">• Interface• Storage• Package• Process• Command• File	<pre>type: classes:: "Promiser" ---> {"Promisee1", "Promisee2"} attribute_1 => value 1, attribute_2 => value 2, attribute_3 => value 3,</pre>

Figure 3.3: Promise types and grammatical form of a Cfengine 3 Promise

The type part of the above syntax describes the subject of the Promise. For the purpose of illustration, let us assume the promise is of type file. The class section of the syntax presents the scope of the Promise. In other words, the class answers the question of when and where the promise is going to be applied such as on all computers or else only on those computers that are running Linux as their operating systems. In addition to the possibility of using user defined classes for scoping of promises, Cfengine 3 has a set of built-in classes that could be used for the same purpose.

Very often, the Promisee part of the syntax above is not used in Cfengine 3 except for the purpose of documentation. The very significant part of this line is the Promiser: the actual entity that is planning to apply the Promise. The file "passwd" could be a good example here. In the body part of the Promise for example, the value of 644 could be assigned for permission attribute of passwd file. That means, the promiser passwd file will promise to have that specific value for its permission attribute.

As a Promise Theory based configuration management tool Cfengine 3 plays the role of implementing the Promises made by each configuration item

to produce a policy conformant computer system. Even after the implementation, assuring the continuous conformance of a system with its specification is the responsibility of Cfengine 3. The task of continuous maintenance or change management is claimed to be easier for Cfengine 3 due to its underlying modeling approach.

In other words we want to be able to promise that the system is correct, verify this and only make changes if our promises are not kept. If you want to think ITIL, think of this as a service that Cfengine provides. [17]

Cfengine 3, as a Promise Theory inspired configuration management tool is therefore responsible for continuous check up of the state of each configuration item against its respective Promises to assure a continuously policy conformant computer system in addition to the initial implementation of those Promises.

3.4 Knowledge Management and Promise Theory

The idea of intertwining Knowledge Management with Promise Theory is coming to the scene with a nice goal of having Semantic approach towards modeling the complex information within a Policy based management framework.

This work shows that there is a two-way mapping between promises and topic maps that enables a simple formal representation of human understanding to be codified[21].

By using Cfengine 3 as a mediating factor, the special relationship between Topic Maps and Promise Theory has been stated by [21] in such a way that Topic Maps will be used to represent the knowledge within a Promise Theory based configuration domain; conversely, Cfengine 3 will be used to codify the knowledge represented by topic maps as a Promise of having a given semantic structure.

The reasoning may be summarized as follows: a simple knowledge model can be used to represent a simple policy configuration model; conversely, a simple model of policy configuration can represent indeed manufacture a knowledge structure, and there is a natural promise engine that can implement this map-ping: cfengine [21].

The fact that Topic Maps are better than other knowledge representation approaches for structuring of information that will be used by human beings has been discussed in a previous section. In addition, the appropriateness of Promise Theory for modeling the autonomy of agents in a Policy based management framework was also discussed section 3.3. Therefore in this project, the combination of these approaches is believed to be the best choice for the

task of enabling an integrated knowledge management within a Policy based management framework. Interesting similarities between the two approaches are pointed out in the following paragraph.

Both Promise model and Topic Maps have the same basic world view: Principle for reduction of knowledge into atoms and autonomy of concepts that automatically avoids overlap and conflict[21]. A Promise Model revolves around the autonomy of agents. In a similar fashion, Topic Maps reduce knowledge structure to the level of subjects or concepts.

Promise Theory models intent that will produce knowledge after implementation. Configuration requirements are intents that could be modeled and implemented with the help of Cfengine3. The collection of implemented intents in a configuration management domain will make up configuration knowledge which could easily be accessed by non experts. Its ability of modeling intent is a good knowledge management approach which makes Promise Theory preferable for the enhancement of configuration knowledge management.

The use Promises to encode knowledge structure which is represented by using Topic Maps is found to be good in assuring the integrity of configuration knowledge. This is because; Promise Theory is good at change management in such a way that un kept promises could be reported as a change for a timely reaction. In the same way, a promised knowledge structure should be kept in order to have an integrated knowledge management. If that is not the case, Cfengine 3 can make a report letting timely adjustments to ensure the integrity of the whole domain knowledge.

Cfengine 3 classes are also found to be equivalents of Topic Maps scopes. In Topic Maps one can define a scope by listing set of topics that could give a specific context. In the same way, Cfengine 3 has set of built in classes. In addition, Cfengine 3 users are free to define their own classes. That is the other similarity found between the two approaches.

Enabling a graphical navigation of a semantically represented information domain is the goal of unifying Topic Maps with Promise Theory[21]. It is different from a simple database which mainly serves as a repository of information with limited set of associations among the different entity types. It is also far from a hyper link connected set of traditional Web Pages which are more of information locators than information finders.

3.5 Configuration Information

This thesis work is about enabling an integrated knowledge management in the area of configuration management in general and a Promise Theory based configuration management in particular. In chapter two, the meaning of knowledge has been defined as an understanding which is made up of collection of information. Therefore, before dealing with the high level concept of knowledge management, the meaning of the underlying information in this case the meaning of configuration information needs to be precisely defined. That is what this section is about.

3.6. CONFIGURATION MANAGEMENT AND KNOWLEDGE MANAGEMENT: RELATED WORKS

Configuration information of a domain comprises the underlying configuration facts. Configuration parameters are the major types of configuration facts. For this literature, a configuration parameter are defined as follows.

Definition 9 *Configuration parameter: is a property of a computer system that will play a role in determining the configuration status of the whole system.*

File permission set is a good example of configuration parameters. The wrong set of values of a permission of a file could matter the configuration status of the whole computer system. In addition to configuration parameters, relationship between the different components including the dependency relationship among the different packages comprises the configuration information of a computer system.

In a Promise Theory inspired configuration management domain, one can define the meaning of configuration information based on the two kinds of Promise that could be made in such a domain. The first of these kinds is a Promise made by a configuration item to behave like another. This kind of Promise deals with the attributes and behaviors of the item. Therefore, any promise made by a configuration item about the values of its attributes and the way it behaves is part of configuration information.

The second kind of Promise is a Promise to utilize the Promise of another. Association among the different configuration items such as the dependency relationships and service giving and receiving relations belongs to this category of Promises. Finally, the set of all Promises is what makes the configuration information of a Promise Theory based configuration management domain.

3.6 Configuration Management and Knowledge Management: Related Works

The way configuration information organized and stored was a total mess until the recent move to wards the notion of using Configuration Management Database (CMDB). In data centers, with thousands of machines, tracking information about things like files permissions, is a challenging and complex task. That is why an efficient management mechanism for configuration information is believed to be crucial by many others including the Information Technology Infrastructure Library (ITIL). In [22], ITIL is described as a world wide known framework of IT (Information Technology) service management aimed at the provisioning of quality IT services. CMDB is ITIL's initiation for efficient management of organizational wide configuration information.

CMDB is a federated hub for configuration information and it can be described as the heart of ITIL framework. Some ITIL practitioners described it as "the ultimate source of truth". All ITIL processes rely on CMDB to furnish them with trustworthy configuration item (CI) information, relationships, and interdependencies.[23]

3.6. CONFIGURATION MANAGEMENT AND KNOWLEDGE MANAGEMENT: RELATED WORKS

The centralized storage of IT related information in a CMDB will be useful for a further analysis of the underlying information. By using statistical analysis, one could get the over all performance of each configuration item based on the centrally stored information. In addition, the information in a CMDB can easily be used for the development of intelligent systems. Timeliness and quality of information search will also be enhanced by using CMDB. However, although CMDB has a lot to do with solving the problem caused by the lack of integrated knowledge management, it is also claimed to have downsides when the size of information gets bigger. That draw back of CMDB is related to its syntax based approach of information management. That is why ITIL has started a related work on extension of CMDB with a semantic based knowledge management scheme.

Therefore, the restructuring of ITIL implementation through knowledge management frame work is believed to be of great advantage to ITIL not only in terms improving its efficiency, but also in terms of its success as stated in.[23]

As the need for a higher level abstraction of domain knowledge increases, a number of related research works are going on including by [24] on the capability of extending databases using knowledge representation approaches such as Topic Maps.

Topic maps are a new ISO standard for describing knowledge structures and associating them with information resources, a solution for organizing and accessing large and continuously growing information pools.They are dubbed as the 'GPS of the information universe'. One possibility to permanently store Topic Maps is using Relational Databases. In this document I present a possible database schema that allows to store Topic Maps[24].

System maintenance as part of configuration management is the process of detecting and correcting system anomalous behaviors. For anomaly detection systems such as Cfengine, managing a multitude of reported events about the status of each involved host was a serious problem observed for a number of years. Recently however, [25] has discovered the possibility of using knowledge representation technologies to solve this problem. In a related work by [26], a prototype for an ontology based creation and management of Intrusion detection systems has been presented and claimed to be feasible.

The possibility of automating configuration of Internet Protocol(IP) networks by using a semantic technology is also found to be promising as stated in [27]. A related endeavor of creating an ontology based sensor network environment was also reported to be viable in [28].

Generally speaking, most of the related works reviewed for this literature have used RDF as a knowledge representation approach and have claimed to be successful especially in using its reasoning feature. To the author's understanding, one of the possible reasons behind this could be the fact that most of

3.7. CURRENT STATUS OF CONFIGURATION KNOWLEDGE MANAGEMENT

them were focused on the representation of knowledge from the perspective of machines. The other reason could be the fact that RDF is old enough to be known compared to the other semantic technology giant, namely Topic Maps.

Although it is common for almost all knowledge representation tasks, incompleteness of ontology was a problem observed in most of the related literatures. Finally, the author couldn't find any research work done on the enhancement of configuration knowledge management based on semantic technologies.

3.7 Current status of configuration knowledge management

Most of the organizations today have no systematic way of managing their configuration information. That means, configuration information such as IP address of a machine and its network service status are stored in a scattered and unorganized manner. Some monitoring tools such as Nagios have done some work in the presentation of monitoring information from different sources in an organized manner.

The idea of using CMDB is also a great work done so far on solving the above problem. However, although CMDB is a great work done so far and also a very important aspect of integrated knowledge management, the need of having a higher level knowledge management approach in addition to CMDB has been stated in different literatures including [23]

CMDB has been automated in many infrastructure management systems to collect configuration data from different assets. The existence of CMDB may present the chimera that all the required information is available in the repository. However, the crux of the issue is the degree of 'find-ability' for these data within the existing federated database.

This new way of enhancing knowledge management efficiency in the area of configuration management, requires the possibility of automatic data extraction from previous storage locations and possibility of automatic updating of permanent storages such as the CMDB in order to increase the quality of the underlying information. The possibility of those two features namely, automatic data extraction and also an automatic updating of the stored information were proved by ITIL.

This thesis work is about showing the possibility of having an integrated configuration knowledge management by the combined effort of Topic Maps and Promise Theory. As pointed in figure 1.1, the structure of configuration information that is stored in a CMDB and other documents will be represented according to the standards of Topic Maps. That domain knowledge structure will be implemented as a Promise of knowledge structure by using Cfengine 3.

Chapter 4

Design and Implementation

This chapter presents the heuristic Methodology used in the restructuring of a Promise Theory based Configuration information according to the standards of Topic Maps. Following the short description of the Start up phase, a Conceptual model is presented to show the basic ideas behind the domain under consideration.

The result of the Physical design phase; domain ontology shows the basic concept and associations that exist within the domain. As stated in different literatures, ontology design is non trivial. The reason behind this difficulty could be the fact that ontology is about modeling of human domain knowledge which is too complicated to represent fully. A discussion on the tool used for implementation and a sample snap shoot taken after the implementation is presented in the final sections of the chapter.

4.1 Methodology

The lack of formal Methodology for Topic Maps based information structuring is the reason behind the development and use of a heuristic Methodology in this thesis work. Although they are found to be complex especially for Topic Map beginners as to the authors understanding, the two promising Methodologies proposed by [9] and [29] have been considered as alternatives before the decision of using a heuristic Methodology.

The heuristic Methodology was designed based on the good features of the Methodology proposed by [29] as well as the Methodologies used in Information Architecture and Database design. The five phases of this Methodology are:

- Start up
- Conceptual design
- Physical design
- Implementation
- Evaluation.

4.2. PROJECT START UP PHASE

The purpose of Start up phase is to create a clear understanding of the project by the Topic Map developer and the project stack holders. In this phase,

- the project stack holders will be identified
- the scope of the project will be determined
- the vision of the project will be clearly set
- potential data sources will be identified
- the tools and approaches that will be used in the project will be identified

The Conceptual design is aimed at enabling the Topic Map developer gets clear understanding of the domain under consideration. The result of this phase is a conceptual model that is acceptable by the project stake holders with out any detail of underlying knowledge representation approach. The idea of System Thinking could be used in this phase in order to identify the inputs, process, outputs and goal of a specific domain.

The Physical design phase will result in a model that represents the domain based on the underlying knowledge representation technology: Topic Maps. Mostly, the result of a Physical design phase is a domain ontology as it is in this present work. Codifying the designed ontology and populating the resulting Topic Maps with real data are the activities of Implementation phase.

Finally, the resulting Topic Map could be evaluated based on the expected functional requirements or based on its acceptability by the project stake holders in the Evaluation phase of the Methodology as presented in the following chapter.

4.2 Project Start up phase

The vision of this thesis work is to show the possibility of solving the problem of configuration knowledge management by using Topic Maps and Promise Theory. The combined effort of the two approaches is tested if it yields the expected result of easier configuration knowledge management.

The project is about simplifying configuration knowledge management from the perspective of humans. Therefore, experts of system administration and people working in the area of a Promise Theory based configuration management are considered to be the stake holders of this project.

The effort that was made to reuse previously designed domain ontology was unsuccessful. Therefore, literatures written on the area of Promise Theory and configuration management were used as sources of information. However, the main data source used in the identification of domain concepts and association among those concepts was the reference manual of Cfengine 3. That is because, it was found to be rich in covering all major aspects of the domain under consideration.

Until recently, Unified Modeling Language (UML) was the tool used for documenting most ontology designs. Currently however, Onotoa is becoming

4.3. A CONCEPTUAL MODEL FOR A PROMISE THEORY BASED CONFIGURATION MANAGEMENT

the tool that could be used to define Topic Maps schemas in a full detail. It is a completely ISO standard compliant that has almost every features required to define the schema of Topic Maps. In this Project, Onotoa 1.0.2 is used for documenting the resulting ontology designed for the domain which is under consideration.

In this thesis work, Topic Maps is used to design the ontology for the underlying domain knowledge. Cfknow: the knowledge modeling component of Cfengine 3 on the other hand is used to encode the designed ontology according to the Promise Theory modeling approach.

4.3 A Conceptual Model for a Promise Theory based Configuration Management

The purpose of this phase is to let the Topic Map developer get to know the domain of a Promise Theory based configuration management. The idea of System Thinking is used as an approach to understand and model the domain under consideration.

An approach is a way of going about tackling a problem, and obviously a particular approach may be relevant to more than one subject, just as an experimental approach might be taken to the problem of physics, psychology, agricultural, and many other subjects. 'A system approach', however, although it conveys the idea of a method of attack, does not readily convey to most people much idea about the content of the method [30].

In this literature, a system is a set of interrelated components that can take input, conducts processes to give some sort of output that will enable the system to achieve its goal. Taking a Promise Theory based configuration management domain as a system, the idea of System Thinking is used to identify its inputs, processes and outputs in order to have a conceptual model of the respective system. The idea behind this is the fact that a domain could better be understood by knowing what it takes in as input and what process it performs to give the desired kind of result that will assure the achievement of its over goal.

4.3.1 Inputs of a Promise Theory based Configuration Management domain

As the result of the analysis conducted, a high level configuration policy is found to be the input for a Promise Theory based configuration management system. The high level configuration policy specifies the desired kind of system expected as the result of the configuration process. Most of the time, this high level specification is made by a group of people including the system administration and managerial personnels of an organization. At this stage, the concern is on the behavior of the expected system rather than the underlying technical detail.

4.3. A CONCEPTUAL MODEL FOR A PROMISE THEORY BASED CONFIGURATION MANAGEMENT

The kind and size of network desired by an organization is a very good example of inputs to the system under consideration. It can be stated as a medium size local area network. The technologies required to realize this desire will be identified in the processes will follow the identification of inputs. In addition to this example, the following list of things could be forwarded as inputs for a Promise Theory based configuration management domain.

- The required number and type of machines
- Operating System architecture of each machine
- Packages that need to be installed on each machine
- The services that should be given by each host
- Storage related issues such as the type of file system
- Security issues including user management aspects

4.3.2 Processes of a Promise Theory based Configuration Management domain

The high level configuration policies that serves as inputs will go through two major processes. The first one is low level configuration specification. This process depends on the architectural information such as the device detail and the desired type of topology of a system. For example, some software couldn't be installed on some kinds of computers. Therefore, the identification of device details will help the system administration personnel in the process of low level configuration specification. The result of this process is a set of promises made by each major configuration items about their attributes and relationships with other entities.

The Promise Theory inspired configuration tool will handle the second kind of process or the process of implementation. That means, Cfengine 3 will enforce the promises made by each of the configuration items. This will make each configuration item to behave according to its promises. In addition to initial implementation, Cfengine 3 is also responsible for the following task of change management. In managing changes, Cfengine 3 will take corrective action in addition to playing the role of informing concerned parties about promises that are not kept.

4.3.3 Out puts of a Promise Theory based Configuration Management domain

The output of these processes is a set of computers with the proper configuration values for their configuration parameters (promised values of their attributes). In addition to their own attribute values and way of behaving, a set of Promises about the right kind of relationships among those configuration items is expected as the result of those processes. According to Promise Theory, this is the policy or desired state of computer systems. This out put of

4.3. A CONCEPTUAL MODEL FOR A PROMISE THEORY BASED CONFIGURATION MANAGEMENT

policy conformant system will enable the system to serve its intended purpose at an acceptable level of performance as the goal of a Promise Theory based configuration management systems.

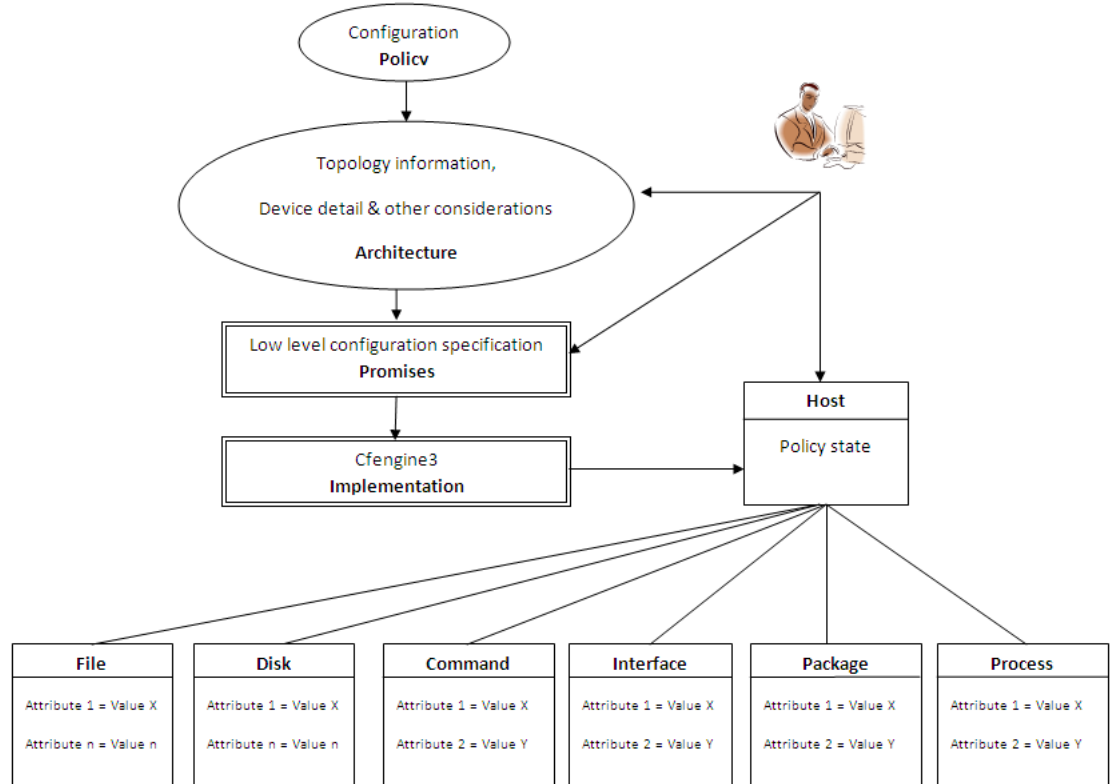


Figure 4.1: A conceptual model of a Promise Theory based configuration management domain.

Figure 4.1 is a summary of the discussion presented in this section. As shown in the figure, the domain of a Promise Theory based configuration management takes high level configuration policy as its input. The high level policy will be broken into low level configuration specifications or what we call it promises in a Promise Theory modeling approach.

The Promise Theory inspired configuration management tool, Cfengine 3 handles the process of implementation. The out put as shown in the figure is a policy conformant computer system with each of its configuration items having the proper attribute-value pairs. Files, packages, disks, process, service and interface are the major configuration items found from the reference manual of Cfengine 3. The right promise made and kept by all or most of this configuration items gives a host or computer system which conforms to its policy specification. Generally speaking, the goal of a Promise Theory based configuration management is to let computer systems serve their intended purpose at an acceptable level of performance.

4.4 Domain ontology for a Promise Theory based Configuration Management

As mentioned in [25], some researchers have categorized the different kinds of ontologies as Task ontology, Method ontology, Application ontology and Domain ontology: ontology defined for conceptualizing the particular domain such as a Promise Theory based configuration management domain.

This section discusses the physical design phase of the Methodology adopted for this work. The resulting domain ontology presents the basic concepts and association among those concepts of a Promise Theory based configuration management. The identification and definition of the concepts and associations was made according to the standards of Topic Maps.

The domain ontologies formalize the available knowledge about the domain of the future application, so all software automation approaches will use them[31].

The design of domain ontology has many advantages. Firstly, it is the major part of knowledge representation for an enhanced knowledge management scheme. The design of ontology also helps experts share information in a precise and integrated manner. As quoted above, the design of domain ontology is also very important for intelligent systems that can make use of domain information.

Since Topic Maps is quite rich in possibilities, the design of Topic Maps ontology in this work is based on the minimum required features as recommended by [32]. Accordingly, the identification and definition of topic types, occurrence types and association types has comprised this phase of the project.

4.4.1 Topic Types

According to the principles of Topic Maps information restructuring, any subject of interest is represented by a topic defined for its representation. Topics with similar characteristics will form a topic type. There is no strict rule about the degree of similarity expected of the topics with in a specific topic type. Therefore, one can categorize set of topics to form a topic type given that they have one or more characteristics in common.

This section presents the major topic types that exist in a Promise Theory based configuration management domain as shown in figure 2.2. The meaning of each topic type according to this literature is also part of the section. Names of topic types are made to be singular with an initial letter capitalized to follow the formality recommended by some Topic Map experts.

The result of the conceptual design phase is used as the basis for the identification of existing topic types. As shown in figure 4.1, person is one of the topic types identified and defined to include any person that may play one or more roles in the specified domain. For example, a person could be a user or administrator of the system. The person Eskedar could be an example

4.4. DOMAIN ONTOLOGY FOR A PROMISE THEORY BASED CONFIGURATION MANAGEMENT

topic of type person who can use any of the machines in the specified domain. Therefore, any person who plays one or more roles in the domain which is under consideration is supposed to be an instance of this topic type.

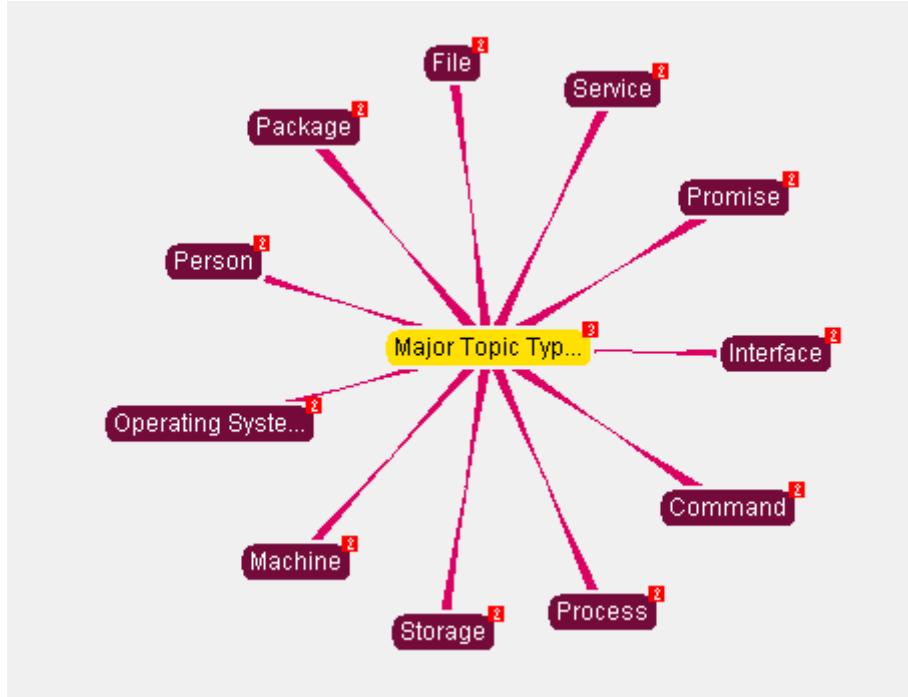


Figure 4.2: Topic Types of a Promise Theory based configuration management domain

Machine is any device that can accept and process information to provide a desired result based on a program or sequence of instructions on how the data is to be processed. Computers are the main types of machines for this thesis work. That is why the two terminologies are interchangeably used in different parts of this literature. Personal computers, laptops, server computers all belong to this topic type.

A computer runs one or more operating systems. Operating System is a program that bridges the gap between the package and the underlying hardware components of a computer. Operating system as a topic type consists of different operating systems as topics including Windows XP, OS X, and Linux. Packages on the other hand are application programs designed to serve specific purpose. It could be to give a network service or web service on a machine. The package named Apache is a good example topic of type package which is designed to enable a web service.

Service is a specific functionality of a computer system such as a web service or network service provided by a machine. A process which is defined in this literature as a package or part of a package on action enables the provisioning of services by performing real activities behind the scene. A command is a utility that can be used by users to start a specific process if there is no schedule set for the same purpose. A very good example that can illustrate

4.4. DOMAIN ONTOLOGY FOR A PROMISE THEORY BASED CONFIGURATION MANAGEMENT

the relation ship among those topic types is that of a web service that needs a process such as "httpd" running on the background which is started by a command:"httpd start".

Storage is a logical partition of a physical storage media that mainly serves as a location for the different kinds of files. [33] has defined a file as a complete, named collection of information such as a program, set of data used by a program or a user created document. Interface is defined by [33] as a device used to provide network access to a computer. The definitions given by [33] fits the meaning of those topic types in this present literature.

Although not a complete one, the major topic types of a Promise Theory based configuration management model is presented and briefly described. Most of those topic types are found to be the same as the major promise types of Cfengine 3 as presentenced in figure 3.3. This is one of the interesting similarities observed between a Promise Theory based models and a Topic Maps based information structuring. In other words, promise types of a Promise Theory based model are found to be equivalents of Topic Maps topic types.

4.4.2 Occurrence Types

As stated in section 2.4.2, an occurrence type could be a property (internal occurrence) of a topic or could be an external information resource (external occurrence) such as a web page that has relevance with the topic. There is no formal procedure to follow for the identification of Topic Map occurrence types.

In this project, we have used the list of LVAL(Left value)found from Cfengine 3 reference manual as the internal occurrence types of a Promise Theory based configuration model. LVAL (Left value) is a collective name given to the constraints that are determining the behavior of configuration items in a Promise Theory based modeling approach. The value of each LVAL is determined by another value called RVAL (Right value).

Configuration management is the process of constraining the behavior of a network of machines so that each machine's behavior conforms to predefined policies and guidelines and accomplishes predetermined business objectives[16].

Rather than doing it by unsystematic way of listing out the assumed properties of each topic type, we preferred to make it systematic by using an already existing list of properties (internal occurrences) of each topic type. That is why the list of Cfengine LVALs is chosen to be basic source from which we have gathered and defined the occurrence types of the domain under consideration.

However, there also some internal occurrences such as address of a person those are not included in the list of Cfengine LVALs. That is why some internal occurrences and also all external occurrences are added to the list based on the authors belief that each of the added occurrence types are required as information sources of the resulting domain knowledge.

4.4. DOMAIN ONTOLOGY FOR A PROMISE THEORY BASED CONFIGURATION MANAGEMENT

One can list out all the properties of a topic as internal occurrences. The problem however is that as the size of the domain information increases, the less related properties of a topic will only create information over load leading to a less efficient knowledge management. For example, gender as a property of a person is less relevant information to be stored as part of the underlying domain knowledge.

Although not a complete one, figure 4.3 has presented most of the occurrence types identified for a Promise Theory based configuration management domain. However, a further and iterative group discussion with the project stake holder is expected to result in a better list of occurrence types for the domain under consideration.

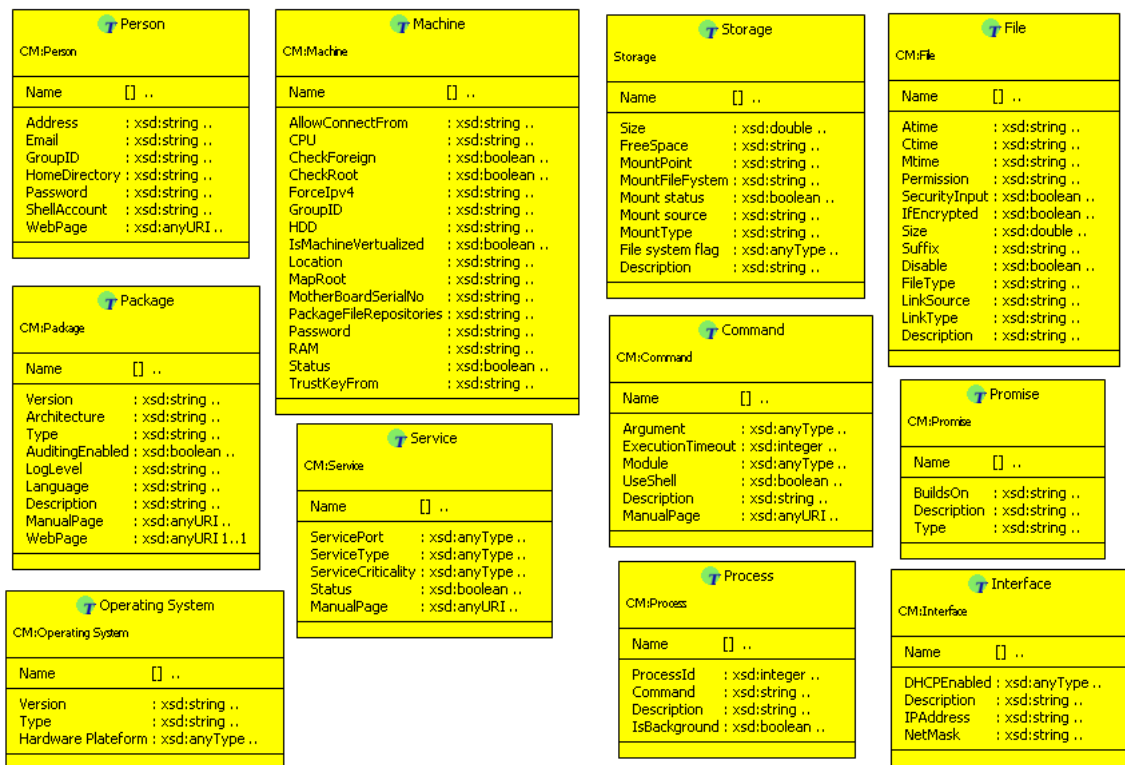


Figure 4.3: Occurrence Types of a Promise Theory based configuration management domain

The above figure presents the list of occurrence types identified in relation to each of the previously presented topic types that are found in the domain under consideration. The figure is a screen shoot taken from the Topic Maps ontology editor which is used for this thesis work namely, Onotoa. The presented boxes and list of properties should not be confused with an entity relationship model of a Relational database. However, it can be used as a mediator in the process of mapping the designed Topic Maps to a Relational database such as CMDB of a Promise Theory based configuration management domain. Each of the boxes or topic types could be mapped to the different tables of a Relational database schema. The occurrence types can directly be mapped to

4.4. DOMAIN ONTOLOGY FOR A PROMISE THEORY BASED CONFIGURATION MANAGEMENT

the fields of those tables. In addition, the data types could be used in the same way as the data types of the respective database fields.

Most of the occurrence types have the same description as their generic meaning. For example, CPU as an occurrence type gives information about the CPU model and speed of the respective machine. Therefore, one can easily deduce the kind of information that will be gained from each of the well known occurrence types. However, there are occurrence types that might need description such as the check foreign occurrence type of a machine. The following table has the description of those occurrences that are assumed to be ambiguous for the readers of this documentation.

Definition of occurrence types that are related with the topic type machine are listed here below.

Definition 10 *AllowConnectFrom*: list of IPs or hostnames that may have more than one connection to a server port.

Definition 11 *CheckForeign*: tells whether there is a need for checking permissions on the root directory during depth search.

Definition 12 *CheckRoot*: list of host names or IP addresses to grant full read-privilege on a server

Definition 13 *ForceIpv4*: tells whether there is a forced use of IPv4 in connection

Definition 14 *IsMachineVirtualized*: tells whether a machine is virtual or not.

Definition 15 *PackageFileRepository*: a list of machine-local directories to search for packages

Definition 16 *TrustKeyFrom*: list of IPs or hostnames from whom a machine will accept public keys on trust

Definition of occurrence types that are related with the topic type storage are listed here below. A storage is a logical partition of a physical storage medium.

Definition 17 *FreeSpace*: absolute or percentage minimum disk space that should be available on a storage before warning.

Definition 18 *MountType:Protocol* type of remote file system

Definition 19 *FileSystemFlag*: List of menu options for *bsd* file system flags to set

Definition of occurrence types that are related with the topic File are listed below.

Definition 20 *Atime*: Range of access times (*atime*) for acceptable files

Definition 21 *SecureInput*: shows whether input files are writable by unauthorized users.

4.4. DOMAIN ONTOLOGY FOR A PROMISE THEORY BASED CONFIGURATION MANAGEMENT

Definition 22 *MoutType*: Menu option for type of links to use when copying such as symlink and hardlink.

Package related occurrence types are defined here below.

Definition 23 *AuditingEnabled*: shows whether log auditing feature of a package is enabled or not.

Definition 24 *LogLevel*: the reporting level sent to syslog.

Definition 25 *Architecture*: the architecture for package selection such as "x86_64".

The rest of the definition for the occurrence types of the topic types: promise, person, and command are presented here below.

Definition 26 *BuildsOn*: a list of promise bundles that a promise builds on or depends on somehow (for knowledge management).

Definition 27 *HomeDirectory*: is a directory which contains the personal files of a computer user.

Definition 28 *ShellAccount*: is a personal account that gives a user access to a Unix shell.

Definition 29 *UseShell*: shows if a command is embedded in a shell environment.

Definition 30 *Module*: shows whether to expect the cfengine module protocol.

4.4.3 Association Types

Although different authors including Aristotle and Kant have tried to put a clear definition to the different kinds of relationships as stated in [34], identification and definition of relationships has remained to be the complex part of most ontology designs. Even though it lacks a systematic approach and coverage of all possible relationships, this section has presented the major association types that could exist in a Promise theory based configuration management domain. The association types identified and presented in figure 4.4 are defined as here below.

- **Caused By**: an association type where one topic plays the role of affecting the other by changing its state from state X to state Y.
- **Configured By**: when an entity X can bring the necessary changes for entity Y in order to let it serve its intended purpose. The relationship between a machine and a configuration management package such as Cfengine 3 is a typical example of such kinds of relationships.
- **Edited By**: writings such as a file could be amended by an editor such as user of the file for some purpose.

4.4. DOMAIN ONTOLOGY FOR A PROMISE THEORY BASED CONFIGURATION MANAGEMENT

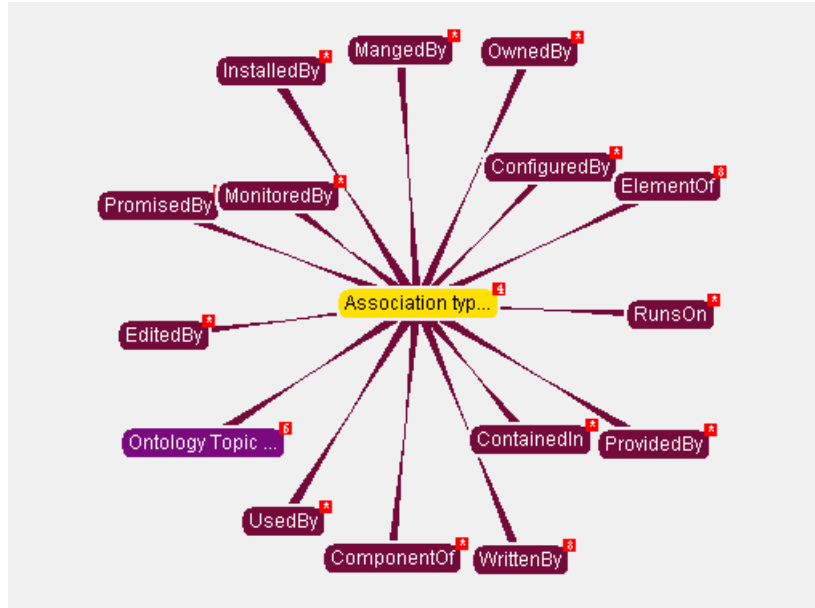


Figure 4.4: Major Associations Types in a Promise Theory based configuration management domain.

- **Managed By:** If X is responsible for the proper functioning of Y in order to let it accomplish its intended purpose, X is said to be the manager of Y. The act of managing includes a continuous supervision and action taking when there is a deviation from the desired way of behaving.
- **Monitoring By:** Monitoring is mainly the responsibility of keeping an eye on some thing. If an entity X monitors Y, it keeps an eye on Y so that it may report to the manager of Y when there is a deviation from the desired behavior.
- **Component Of:** The relationship between an entity X which is a component for another entity Y which plays the role of being a complex for the component X.
- **Owned By:** Ownership is a kind of relationship that could exist between an entity which has become a possession and its owner. The relationship between a file and its owner is an example of this type of associations.
- **Promised By:** the relationship between a promise and its promiser.
- **Required By:** If X depends on Y in order to serve its intended purpose, the relationship between them is called RequiredBy.
- **Runs On:** If entity X performs its activities or execution on entity Y, the relationship is called as RunsOn.
- **Written By:** the relationship between a writing X and its writer Y.

4.4. DOMAIN ONTOLOGY FOR A PROMISE THEORY BASED CONFIGURATION MANAGEMENT

- **Used By:** If an entity X uses entity Y at any point in time for any of its purposes, the the type of association is called UsedBy.
- **Started By:** When an entity X makes another entity Y to start doing something or behave in some manner, the association is said to be of type StartedBy.
- **Provided By:** When an entity X has the potential and willingness of providing some thing to some other entity, the relationship that exists between them is called ProvidedBy.
- **Installed By:** An entity X can put program Y into a machine in order to let it serve its intended purpose. This relation ship between a program and an entity who put it into a machine is called as InstalledBy.
- **Contained In:** If an entity X is contained with in Y either physically or conceptually, the relationship is called as ContainedIn.

4.4.4 Domain Ontology

The result of the physical design phase; ontology of a Promise Theory based configuration management domain shows the major concepts and the relationships among those concepts as shown in figure 4.5. The major concepts or topic types of the underlying domain are person, promise, machine, operating system, package, interface, process, command, service, storage and file. The lines connecting those topic types represent the possible associations that could exist among those topic types. For example, as shown in one of the diagrams, a topic of type interface can have the relationship of being a component of a machine.

The designed domain ontology is not complete. The missing of some topics, associations and occurrence types is one of the reasons behind its incompleteness. In addition, it lacks the definition of association role types and scopes in order to be considered as a complete domain ontology. Time limitation was one of the reasons behind this drawback. In addition, a successful domain ontology design needs a repeated meeting of the ontology designer with the project stakeholders in order to get an acceptable representation of the specific domain knowledge. Although we have put an effort into the option of reusing existing ontology, it was not possible as we couldn't find any previous ontology designed for the specified domain.

Ontologies capture domain knowledge in a generic way and provide a commonly agreed understanding, and as such, are becoming a crucial element of the Semantic Web. Developing and deploying large-scale ontology solutions represents a considerable challenge in terms of the amount of time and effort required to construct the ontology. These challenges can be addressed by the reuse and adaptation of existing ontologie.[35].

4.4. DOMAIN ONTOLOGY FOR A PROMISE THEORY BASED CONFIGURATION MANAGEMENT

Despite its drawbacks, the designed domain ontology could serve as a mile stone for those who would like to use it for designing a complete ontology. In addition, even with its current status, the designed ontology is capable of conceptualizing the domain under consideration to serve the purpose of proofing the hypothesis of this thesis work; enabling an efficient knowledge management by the combined effort of Topic Maps technology and Promise Theory modeling approach.

4.4. DOMAIN ONTOLOGY FOR A PROMISE THEORY BASED CONFIGURATION MANAGEMENT

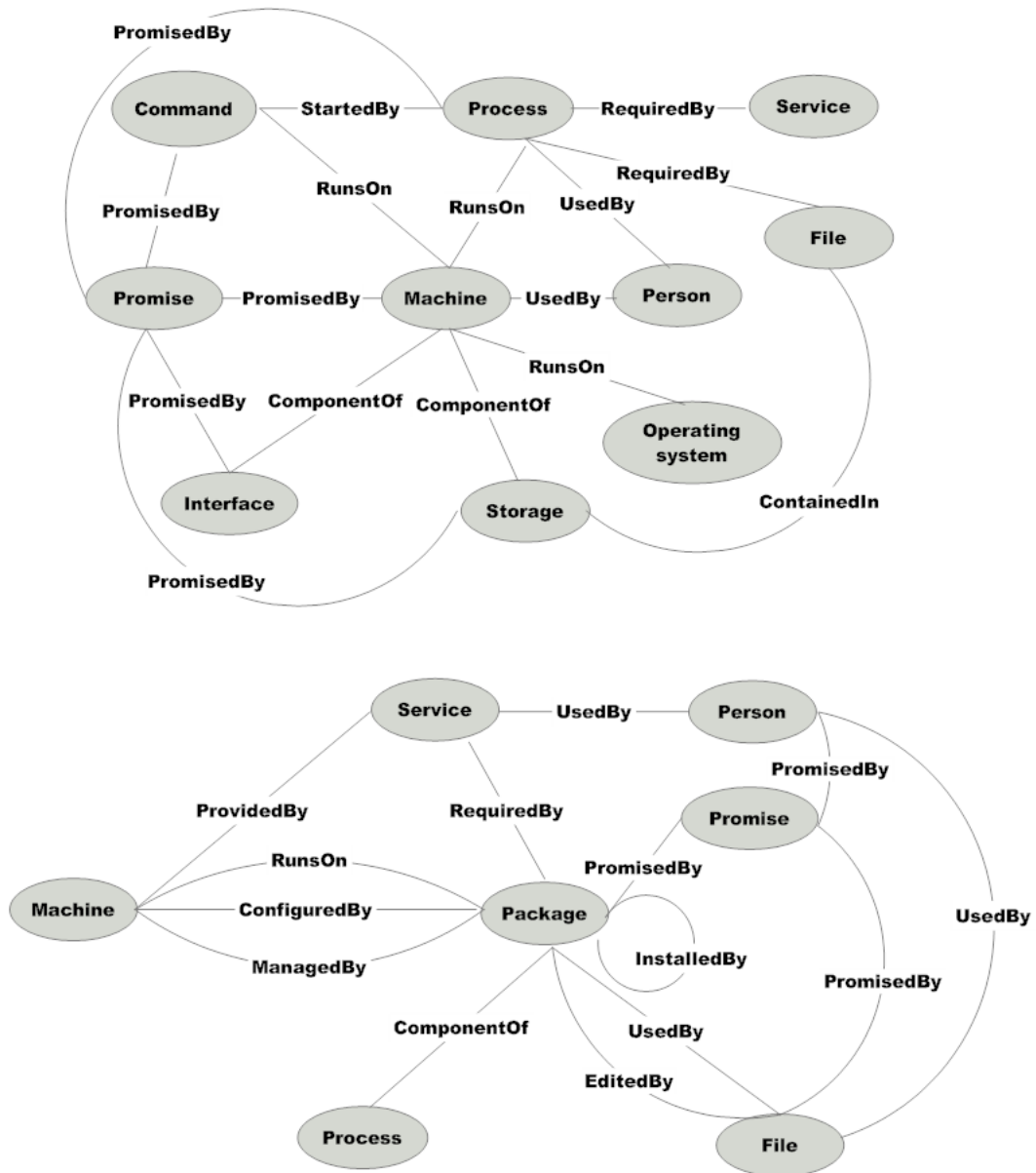


Figure 4.5: Ontology of a Promise Theory based configuration management domain

4.5 Implementation of the ontology on Cfengine 3

There are dozens of available Topic Maps development tools. In this thesis work however, Cf-know: a knowledge modeling component of Cfengine 3 is used to implement the designed Ontology. Cf-know is chosen because, it is the only tool that can serve as a bridge between the two approaches namely, Topic Maps and Promise Theory modeling approach. The ontology which is designed according the standard of Topic Maps is there fore encode to form a knowledge structure by the help of Cf-know. The reader is referred to the previous section for the ontology designed for the domain under consideration.

Cfengine's knowledge agent cf-know is not a generic topic map language: it provides a configuration language for managing a knowledge base that can be compiled into a topic map[21].

Cfknow has an editor that enables users create and modify their Topic Maps by editing its textual notation. It also has a browser that helps users to navigate through the underlying domain information. The graphical visualization feature supported by Cfknow is also believed to play a significant role in simplifying the ability of viewing the relationship among things.

The lack of formal query language and a form-based interface to simplify the task of implementation are the two drawbacks observed in Cfknow. In addition, Cfknow's capability of consistency checking was not seriously tested. However, its capability of checking weather promises are kept or not is found to be similar to the function of consistency checking. In Cfknow, knowledge structure is promised at the time of implementation. If things run out of their promised structure, Cfknow will report the existence of inconsistency by stating the fact that a promise on knowledge structure is not kept. This is a great role that could be played by Cfknow in keeping the integrity of underlying domain knowledge.

The diagram below shows a sample information search conducted on the implemented ontology with a key word of Service1. The port used by and the explanation given to Service1 are presented as shown in the figure. Topics that have association with the service in question are also presented in the association section of the resulting page. For example, "Apache" is a package required by that specific service. Topics of the same type such as Service2 and Service3 are also presented so that the user can go for any of those topics if interested .

Service 1

Service::Service1

This topic "[Service1](#)" has type "[Service](#)" in map version 1.0

Occurrences of this topic:

- [Explanation](#), :
"Web service" (Text)
- [Port No](#), , :
"80" (Text)

Associated with this:

- Service1 "[provided by](#)"
 - [PCA](#)
- Service1 "[requires](#)"
 - [Apache](#)
- Service1 "[depends on](#)"
 - [httpd](#)
- Service1 "[has promised](#)"
 - [Promise5](#)

Other topics of type [Service](#):

- [Service2](#) [Service](#)
- [Service3](#) [Service](#)

Figure 4.6: Sample information search using Cf-know

4.5. IMPLEMENTATION OF THE ONTOLOGY ON CFENGINE 3

The screen shoot presented below shows part of the Cfknew source code that produces the above result.

```
#####
# The Map
#####

bundle knowledge promise_topics

{
vars:
  "impl[f]" string => "implements";
  "impl[b]" string => "is implemented by";

  topics:
#####
# Topics are foundation
#####

Package::
  "Apache"      comment => "Web server";

Service::
  "Service1"    comment => "Web service",
                association => a("provided by","PCA","provides");
  "Service1"    association =>a("requires","Apache","required by");

Process::
  "httpd"       association => a("is required by", "Service1", "depends on");

Promise::
  "Promise5"    comment => "Promise made by Service 1",
                association => a("is promised by","Service1","has promised");

  occurrences:
#####
  Service1::
    "80"
    represents => { "Port No", "" },
    representation => "literal";
}
```

Figure 4.7: Part of Cfknew source code

4.5.1 Cfknew and LTM

The Linear Topic Map Notation (LTM), is a compact topic map syntax developed by Ontopia. It can be used to represent the constructs in the topic map standard in a similar way as that of Cfknew. In this section, we have presented a brief summary of the syntax used by the two knowledge modeling tools.

The first construct of Topic Maps ontology is that of topic definition. In Cfknow, topics such as Service1 are defined to be of type such as Service with a double colon sign following the definition of topic types as shown in figure 4.8. A topic can have an explanation by using the key word comment as shown in line 2. The phrase on the right side of the comment command shows the explanation given to that specific topic.

Service::

```
"Service1" comment => "Web service",  
  
Association => a ("provided by", "PCA", "provides");
```

Figure 4.8: Part of Cfknow source code

Line 3 shows the way in which associations are defined in Cfknow. On the right side of the key word association, a function with three arguments is presented. The first argument shows the label that will be attached to the topic in question in this case the label that will be given to Service1. The second argument shows the topic which has association with topic in question in this case PCA. The third argument shows the label that will be attached to the second topic (PCA). Cfknow has an already defined set of functions including the one used in the above syntax as presented in figure 4.9.

```
#####  
# Bodies  
#####  
  
body association a(f,name,b)  
  
{  
  forward_relationship => "${f}";  
  backward_relationship => "${b}";  
  associates => { $(name) };  
}  
  
#####  
  
body association A(f,name_list,b)  
  
{  
  forward_relationship => "${f}";  
  backward_relationship => "${b}";  
  associates => { @$(name_list) };  
}
```

Figure 4.9: Part of Cfknow source code

Occurrence definition has the format presented in figure 4.10. It starts by the key word occurrence as shown on the first line. The name of the topic

will be specified followed by a double colon in this case Service1. The value of the occurrence is given in double quotes as shown on the third line. The occurrence that will have this specific value then will be represented on the right side of the key word "represents". The last line shows the type of occurrence. For most occurrence types it is of type literal. In the case of information resources such as web pages however, URL is used as data type of their occurrence.

Occurrences:

Service1::

```
"80"      represents => { "Port No" },  
representation => "literal";
```

Figure 4.10: Part of Cfknow source code

In LTM on the other hand Topic Maps constructs are defined as shown in figure 4.11. The example presented in the figure shows the way in which a topic type of operating system (OS) was defined. The definition of an occurrence type called OSFigurePrint and an association type called RunsOn is also shown. Finally, the way how to define topics or instances of topic types is shown by taking Linux as an example topic of type Operating system. A public subject identifier is attached to each of the definitions presented in the example.

```
/* Topic type */  
[OS = "Operating system" @"http://psi.ontopedia.net/Operating_system"]  
  
/* Occurrence type */  
[OsFingerPrint = "OS Fingerprint" @"http://psi.ontopedia.net/Operating_system"]  
  
/* Association type */  
[runs = "Runs" = "Runs on" /OS @"http://psi.ontopedia.net/Operating_system"]  
  
/* Topic */  
[Linux: OS = "Linux" @"http://psi.ontopedia.net/Linux"]
```

Figure 4.11: Part of LTM source code

The purpose of this section is to show the relationship between Cfknow and one of the well-known Topic Maps syntaxes that is currently in use. The source codes taken from both Cfknow and LTM show that they both works on the representation of Topic Maps construct. One of the differences we have noted between the tow notations is the simplicity of Cfknow to a Topic Maps beginner than LTM. However, LTM also could be better in other aspects such

4.5. IMPLEMENTATION OF THE ONTOLOGY ON CFENGINE 3

as the coverage of large number of knowledge representation features including Topic Maps scoping.

Chapter 5

Evaluation of Results and Discussion

5.1 Evaluation of Results

The goal of this thesis work is to show the possibility of having an integrated knowledge management in the area of configuration management. The domain knowledge has been represented based on the Topic Maps standard and implemented using Cfengine 3. This section is about the evaluation of the resulting work if it gives a legitimate ground for the proved concept.

To the knowledge of the author, there is no comprehensive and global approach for evaluating Topic Map developed for specific domains such as this. However, some authors including [29] have suggested the acceptability of the resulting work by the project stakeholders as the basic criteria or way of evaluating the success of such projects. Following the same approach, the author contacted the author of Cfengine 3; a well known expert in the area of configuration management in general and Promise Theory based configuration management in particular. In addition, five master students of network and system administration from Oslo University College has been asked about the validity of the proved concept.

Evaluating the performance of knowledge management is a difficult task due to its subjective and situational nature. The personal preference of the evaluators and the size of information stored at the time of evaluation are examples of those factors that make the result of such kinds of evaluation less reliable. Nevertheless, the following suggestions were found from the evaluators.

- The resulting system was found to be good at usability especially in places such as big data centers. However, in cases of smaller size information domain, the evaluators commented on the easiness of using other methods of searching such as commands to get specific information.
- Timeliness of information search was also found to be good with a given

condition that there is the proper kind of underlying infrastructure required for processing the search queries at an acceptable speed.

- The system is believed to scale well if there is a way of scoping the information search to get the desired result. That means, in a large sized information domain, a specific information search could result in a large size of search results. This will add more work on the user in getting the specific information. Therefore, with a condition of having additional features such as aggregation and scoping of topics, the system is believed to scale well for larger size of information domain.

Finally in comparison with the current state of the art, almost all of the evaluators agreed on the need of having such kind of semantic information structuring in order to have an integrated knowledge management.

5.2 Discussion

Earlier in this text, the benefits of using Topic Maps approach to increase find ability of information from the perspective of human users has been presented. According to the author, the ability of Cfengine 3 as a knowledge modeling tool and Topic Maps as a knowledge representation approach are among the strong sides observed in the course of this thesis work. The down sides or challenging aspects observed on the contrary are presented here below.

- The use of either scopes or name types for avoiding the possible ambiguity that could happen in relation to topic names was found to be very confusing.
- Since topic map is on its infancy, getting topic map experts as well as related literatures was found to be very difficult.
- The full application of all Topic Maps features was found to be very complex. The effort of choosing the minimum features that can yield in an acceptable topic map based representation was also found to be very confusing.
- Lack of formal query language by the tool used for implementation, namely Cfengine.
- Lack of thorough test due to time constraint
- Lack of standard Methodology for the development and evaluation of Topic maps

Due to size of scope and time limitation problems, incompleteness of the designed ontology has also become one of the downside of this thesis work. Normally, designing a complete ontology is known to be a time taking and probably impossible task that may need an iterative communication among the project stakeholders to reach at a consensus about the acceptability of the

designed domain ontology . In addition, the purpose of this thesis work is to serve as a proof of concept rather than designing a complete knowledge representation of the underlying domain.

The possibility of using databases to the storage and retrieval of information was considered and found to be very important in having a permanent storage for a large size of configuration information. Although databases plays a great role in restructuring of information, their limitation in enabling a semantic representation of knowledge has been outlined by many authors including [23]. That is why the need and possibility of extending databases by using topic maps is underlined by the author of this literature. The extension of databases with Topic Maps is expected to increase the usability of the underlying data through values that will be added such as categorization and contextualization of domain information.

The use of web pages for information restructuring was also considered and found to be less useful in increasing find ability of information. The drawback could better be explained in such a way that a query for a single topic could result in a page with further work of finding the specific subject to be done by the user. That is why some literatures call such systems as information locaters rather than information finders. The user will only get help in knowing where the information is rather than the exact answer for the specific request.

The lack of thorough test on the sample Topic Map due to time scarcity was one of the major limitations of this work. The size of and type of information used during the testing and the number of people who have been asked for suggestion could have been better had there been enough time for it. In addition to testing, the incompleteness of the designed ontology in terms of its building blocks, the necessary documentation for each of those building blocks and other features such as cardinality of the different association was the other limitation of this thesis work.

Finally, this work has attempted to show the possibility of simplifying configuration knowledge management by using the special relationship between Promise Theory and Topic Maps. A Methodology of Topic Map information structuring has been presented by the author as one of the contributions of this thesis work. In addition, the designed domain ontology can be reused by any one who is interested in having a complete ontology of the domain under consideration.

Chapter 6

Conclusions and Future work

6.1 Conclusion

In a policy based management framework, the formulation of configuration policy by using promises will make knowledge management of system configuration easier. Because Promises maps well into the semantic knowledge representation technologies such as Topic Maps. This paper have attempted to prove this concept by using the semantic technology: Topic Maps and the Promise Theory modeling approach. In the course of this work, the configuration knowledge which was represented based on Topic Maps standard was implemented by using the Promise Theory based knowledge modeling tool to show how they work together to achieve an integrated knowledge management.

The resulting knowledge management scheme which is based on the combination of the two specified approaches is expected to:

- simplify information search.
- save time that could be spend on information search
- enable timely reaction towards critical things such as system failures.
- result in the representation of domain knowledge which is highly learnable even by the non experts of the domain under consideration.
- prevent possible knowledge lose that could happen when experienced employees quit their jobs.

The use of the resulting knowledge management scheme as a reasoning aid for the people working on the area of configuration is found to be valid only to an extent. Reasoning is a process of making inference from a body of information. In other word, it is a process of answering the why based on known facts in relation to the matter at hand. The process of reasoning can be aided by a clear representation of the relationship among the different domain concepts. The proved concept has shown that the semantic restructuring of information based on Topic Maps can help people in reasoning by showing the relationship between things. However, to be called as a real reasoning aid,

6.2. FUTURE WORK

the resulting Topic Maps is expected of doing inference based on some sort of logical calculations .

On the course of this work, in addition to Topic Maps, other knowledge representation technologies namely, full text based information retrieval systems such as Google, subject based classification schemes such as Taxonomy and the major semantic technologies including RDF have been considered. The usability of full text search based approaches for increasing find ability of information is found to be rather a disadvantage as stated in [3]. The subject based classifications on the other hand are found to be less efficient in their use for semantic representation of domain knowledge. However, RDF is found to be very similar with Topic Maps except for its machine oriented design in its purpose as opposed to human audiences as it is in this project.

6.2 Future Work

Modeling relationships was a serious problem observed on the course of this work. Although many authors have tried to define the meaning of the possible relationships among different entities, still it remains to be a confusing and challenging part of ontology design. Further work on the area of generic outline on modeling relationships is highly recommended for a better use of semantic knowledge representation approaches including Topic Maps.

The full implementation of the proved concept needs a well organized information repository such as a configuration management data base that will keep track of configuration related information in an efficient manner. Therefore, the development of a configuration management database is highly recommended for the realization of the proved concept.

The use of either scopes or name types for avoiding possible ambiguities in the design of Topic Maps ontology was one of the problems observed. Therefore, clarifying the relationship between those approaches is one of the further research works recommended by the author. In addition, the author recommends a research work on the design of standard Methodologies for the development and evaluation of Topic maps.

Bibliography

- [1] A. Toffler. *Future Shock*. 1975.
- [2] V. Bush. *As we may think*. 1945.
- [3] S. Smolink. *Convergence of Classical Search and Semantic Technologies*. 2007.
- [4] *The Dublin Core Metadata Initiative. Dublin Core Metadata Element Set, Version 1.1*.
- [5] Ontopia's Website. <http://www.ontopia.net/topicmaps/materials-vs-thesauri.html>. Technical report, Last retrieved April 24, 2009.
- [6] F. Antoniou, F. van Harmelen . *A Semantic Web Primer*. 2008.
- [7] S. Pepper. *Ten Theses on Topic Maps and RDF*. 2002. Technical report, <http://www.ontopia.net/topicmaps/materials/rdf.html> (Acceded May 15, 2009).
- [8] J. Tauberer. *What Is RDF*. (2006). Technical report, <http://www.xml.com/pub/a/2001/01/24/rdf.html>. (Acceded May 15, 2009).
- [9] R. Guescini, D. Karabeg and T. Nordeng. *Topic Maps for Polyscopic Structuring of Information*. . *International Journal of Continuing Engineering Education and Life-Long Learning* , 2006 - Vol. 16, No.1/2 pp. 35 - 49, 2006.
- [10] S. Pepper. *The TAO of Topic Maps*. Technical report, <http://www.ontopia.net/topicmaps/materials/tao.html>, (Acceded May 15, 2009).
- [11] L. Garshol. *What Are Topic Maps ?* Technical report, <http://www.xml.com/pub/a/2002/09/11/topicmaps.html> (Acceded May 15, 2009).
- [12] G. Grnmo, S. Pepper. *Towards a general theory of scope* (2002). <http://www.ontopia.net/topicmaps/materials/scope.htm>(Acceded May 7, 2009).
- [13] L. Garshol, G. Moore. *Topic Maps - Data Model(2008)* . Technical report, <http://www.isotopicmaps.org/sam/sam-model/d0e2429>(Acceded May 15, 2009).

BIBLIOGRAPHY

- [14] R. Lachica. Technical report, <http://www.semantic-web.at>(Acceded March 7, 2009).
- [15] M. Burgess. *Principles of Network and System Administration*. 2004.
- [16] M. Burges and A. Couch. *Modeling next generation configuration management tools*. (2006). Proceedings of the 20th conference on Large Installation System Administration, P. 11 - 11.
- [17] Cfengine-AutoReference. *Cfengine reference manual*. (version 3.0.1b1). Technical report, <http://www.cfengine.org/docs/cf3-reference.html> (Acceded April 10, 2009).
- [18] J. Bergstra and M. Burgess. *A static theory of Promises* (2008).
- [19] *An approach to understanding policy based on autonomy and voluntary cooperation*. (2005), Lecture Notes in Computer Science, Volume 3775/2005, P. 97-108.
- [20] M. Burgess and S. Fagernes. *Promise theory - a model of autonomous objects for pervasive computing and swarms* (2006). 2006.
- [21] *Cfengine on Topical Islands*.(2009)., A preceding at Topic Maps Norway 2009 Conference.
- [22] Office of Government Commerce (OGC) ITIL. *Best practice for Service support*. Twelvth impression 2006.
- [23] M.A Mohamed, M.S. Mohamed, K. O'Sullivan and V. Ribiere. *The restructuring of the information technology infrastructure library (ITIL) implementation using knowledge management framework*.(2008). VINE, volume 39, issue 3, P. 315-333.
- [24] L. Mugnaini. Mapping topic maps on relational databases. Technical report, http://www.geocities.com/xtopicmaps/mapping_xtmon_databases.html.
- [25] M. Adaa. *Ontology for host-based anomaly detection*. Master's thesis, 2007.
- [26] S. Hung and D Liu. *A user-oriented ontology-based approach for network intrusion detection*. Computer Standards Interfaces , Volume 30, Issues 1-2, P. 78-88 2008.
- [27] D. Xiao and H. Xu. *A common ontology-based intelligent configuration management model for IP network devices*. Innovative Computing, Information and Control, 2006. ICICIC '06. First International Conference on P. 385-388 ..(2006).
- [28] C. Goodwin and D. Russomanno. *An ontology-based sensor network prototype environment*.(2006). http://www.cs.virginia.edu/ipsn06/WIP/goodwin_1568983444.pdf.
- [29] L. Garshol. *Towards a Methodology for developing Topic Maps*.(2007). Lecture Notes in Computer Science , Volume 4438/2007, p 20-31.

BIBLIOGRAPHY

- [30] *P. Chackland. Systems Thinking, Systems Practice.* 1981.
- [31] *Ten benefits of a Subject-centric Information Architecture for the future of Social Networking.*, Position Paper for the W3C Workshop on the Future of Social Networking, Barcelona 15-16 Jan., 2009.
- [32] *S. Pepper. Lecture notes*, <http://www.ontopedia.net/pepper/slides/ontologies.ppt>(last retrieved march 12, 2009). Technical report.
- [33] *Microsoft. Microsoft Computer Dictionary.* 2002.
- [34] *John F. Sowa. Knowledge Representation.* 2002.
- [35] *V. Cross and A. Pal. An ontology analysis tool.* 2007.